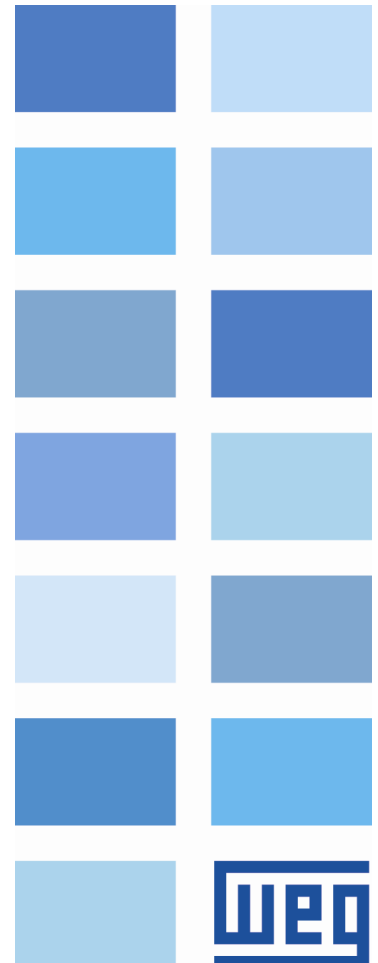


# CANopen

SCW100

## Manual do Usuário





# **Manual do Usuário - CANopen**

Série: SCW100

Idioma: Português

Documento: 10008601693 / 01

Build 221

Data de publicação: 04/2022

A informação abaixo descreve as revisões ocorridas neste manual.

<b>Versão</b>	<b>Revisão</b>	<b>Descrição</b>
-	R00	Cancelado
-	R01	Primeira edição

## SUMÁRIO

<b>SOBRE O MANUAL</b> .....	<b>6</b>
ABREVIACÕES E DEFINIÇÕES .....	6
REPRESENTAÇÃO NUMÉRICA .....	6
DOCUMENTOS .....	6
<b>1 CARACTERÍSTICAS PRINCIPAIS</b> .....	<b>7</b>
<b>2 DESCRIÇÃO DA INTERFACE</b> .....	<b>8</b>
2.1 CARACTERÍSTICAS DA INTERFACE CAN .....	8
2.2 CONECTOR .....	8
2.3 ENDEREÇO .....	9
2.4 TAXA DE COMUNICAÇÃO .....	9
2.5 RESISTOR DE TERMINAÇÃO .....	10
2.6 LEDS DE INDICAÇÃO .....	10
<b>3 INSTALAÇÃO EM REDE CANOPEN</b> .....	<b>11</b>
3.1 TAXA DE COMUNICAÇÃO .....	11
3.2 ENDEREÇO NA REDE CANOPEN .....	11
3.3 RESISTORES DE TERMINAÇÃO .....	11
3.4 CABO .....	11
3.5 LIGAÇÃO NA REDE .....	12
<b>4 S STATUS</b> .....	<b>13</b>
S4 COMUNICAÇÃO .....	13
<b>5 C CONFIGURAÇÕES</b> .....	<b>15</b>
C2 COMUNICAÇÃO .....	15
<b>6 OPERAÇÃO NA REDE CANOPEN</b> .....	<b>16</b>
6.1 ACESSO AO DADOS .....	16
6.2 DADOS CÍCLICOS .....	16
6.3 DADOS ACÍCLICOS .....	16
6.4 OBJETOS RESPONSÁVEIS PELA COMUNICAÇÃO - COB .....	16
6.5 COB-ID .....	17
6.6 ARQUIVO EDS .....	18
<b>7 DICIONÁRIO DE OBJETO</b> .....	<b>19</b>
7.1 ESTRUTURA DO DICIONÁRIO .....	19
7.2 TIPOS DE DADOS .....	19
7.3 COMMUNICATION PROFILE - OBJETOS PARA COMUNICAÇÃO .....	19
7.4 OBJETOS ESPECÍFICOS DO FABRICANTE .....	20
<b>8 DESCRIÇÃO DOS OBJETOS DE COMUNICAÇÃO</b> .....	<b>21</b>
8.1 OBJETOS DE IDENTIFICAÇÃO .....	21
8.1.1 Objeto 1000h - Device Type .....	21
8.1.2 Objeto 1001h - Error Register .....	21
8.1.3 Objeto 1018h - Identity Object .....	22
8.2 SERVICE DATA OBJECTS - SDOS .....	22
8.2.1 Objeto 1200h - Servidor SDO .....	22

---

8.2.2	Funcionamento dos SDOs .....	23
8.3	PROCESS DATA OBJECTS - PDOS .....	24
8.3.1	Objetos Mapeáveis para os PDOs .....	24
8.3.2	PDOs de Recepção .....	25
8.3.3	PDOs de Transmissão .....	26
8.4	SYNCHRONIZATION OBJECT - SYNC .....	28
8.5	NETWORK MANAGEMENT - NMT .....	29
8.5.1	Controle dos Estados do Escravo .....	29
8.5.2	Controle de Erros - Node Guarding .....	31
8.5.3	Controle de Erros - Heartbeat .....	32
8.6	PROCEDIMENTO DE INICIALIZAÇÃO .....	34
<b>9</b>	<b>COLOCAÇÃO EM OPERAÇÃO .....</b>	<b>35</b>
9.1	INSTALAÇÃO DO PRODUTO NA REDE .....	35
9.2	CONFIGURAÇÃO DO EQUIPAMENTO .....	35
9.3	CONFIGURAÇÃO DO MESTRE .....	35
9.4	ESTADO DA COMUNICAÇÃO .....	36
9.5	OPERAÇÃO UTILIZANDO DADOS DE PROCESSO .....	36
9.6	ACESSO AOS PARÂMETROS – MENSAGENS ACÍCLICAS .....	36
<b>10</b>	<b>FALHAS E ALARMES .....</b>	<b>37</b>
	<b>Apêndice A REFERÊNCIA RÁPIDA DOS PARÂMETROS .....</b>	<b>38</b>

## SOBRE O MANUAL

Este manual fornece a descrição necessária para a operação do gerenciador de partidas SCW100 utilizando o protocolo CANopen. Este manual deve ser utilizado em conjunto com o manual do usuário do SCW100.

## ABREVIações E DEFINIções

<b>ASCII</b>	American Standard Code for Information Interchange
<b>CAN</b>	Controller Area Network
<b>CiA</b>	CAN in Automation
<b>CIP</b>	Common Industrial Protocol
<b>CRC</b>	Cycling Redundancy Check
<b>HMI</b>	Human-Machine Interface
<b>ISO</b>	International Organization for Standardization
<b>ODVA</b>	Open DeviceNet Vendor Association
<b>OSI</b>	Open Systems Interconnection
<b>PLC</b>	Programmable Logic Controller
<b>ro</b>	Read only (somente leitura)
<b>rw</b>	Read/write (leitura e escrita)
<b>RTR</b>	Remote Transmission Request

## REPRESENTAÇÃO NUMÉRICA

Números decimais são representados através de dígitos sem sufixo. Números hexadecimais são representados com a letra 'h' depois do número. Números binários são representados com a letra 'b' depois do número.

## DOCUMENTOS

O protocolo CANopen foi desenvolvido baseado nas seguintes especificações e documentos:

<b>Documento</b>	<b>Versão</b>	<b>Fonte</b>
CAN Specification	2.0	CiA
CiA DS 301 CANopen Application Layer and Communication Profile	4.02	CiA
CiA DRP 303-1 Cabling and Connector Pin Assignment	1.1.1	CiA
CiA DSP 303-3 CANopen Indicator Specification	1.0	CiA
CiA DSP 306 Electronic Data Sheet Specification for CANopen	1.1	CiA
CiA DSP 402 Device Profile Drives and Motion Control	2.0	CiA
Planning and Installation Manual - DeviceNet Cable System	PUB00027R1	ODVA

## **1 CARACTERÍSTICAS PRINCIPAIS**

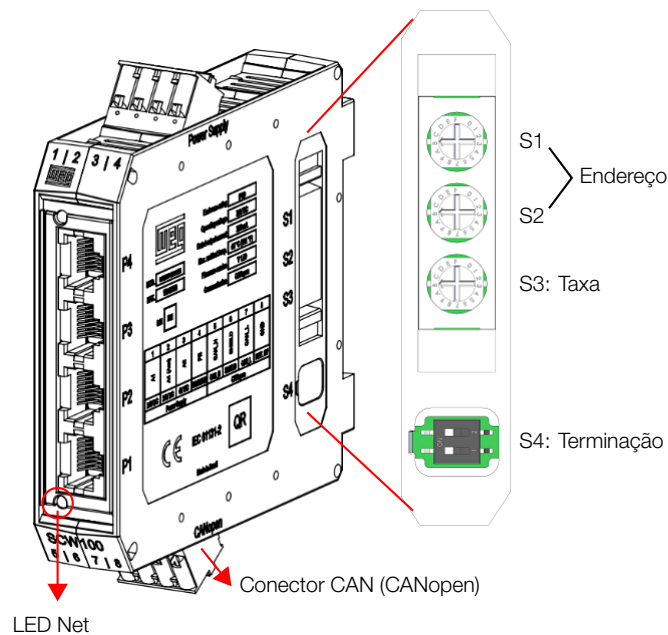
A seguir são listadas as principais características para comunicação com o acessório CANopen do gerenciador de partidas SCW100.

- Serviço de gerenciador da rede (NMT).
- 5 PDOs de transmissão.
- 2 PDOs de recepção.
- Consumidor Heartbeat.
- Produtor Heartbeat.
- Node Guarding.
- Cliente SDO.
- Produtor/consumidor SYNC.
- É fornecido juntamente com arquivo EDS para configuração do mestre da rede.
- Disponibiliza dados acíclicos para parametrização.

## 2 DESCRIÇÃO DA INTERFACE

### 2.1 CARACTERÍSTICAS DA INTERFACE CAN

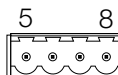
- Interface isolada galvanicamente e com sinal diferencial, conferindo maior robustez contra interferência eletromagnética.
- Alimentação externa de 24 V.
- Permite ao equipamento operar como escravo CANopen.
- Permite comunicação de dados para parametrização e operação do equipamento.
- Possibilita comunicação utilizando taxas de 20 Kbits/s até 1 Mbit/s.
- Comprimento máximo do barramento de 1000 metros.



**Figura 2.1:** Conector, indicações e configurações para o SCW100

### 2.2 CONECTOR

A interface CAN é disponibilizada através de um conector *plug-in* de 4 vias com a seguinte pinagem:



**Tabela 2.1:** Pinagem do conector *plug-in* para CAN

Pino	Nome	Função
5	CAN_H	Sinal de comunicação CAN_H
6	Shield	Blindagem do cabo
7	CAN_L	Sinal de comunicação CAN_L
8	CAN_GND	0V do circuito isolado para interface CAN



## 2.3 ENDEREÇO

O endereço do produto na rede CANopen é configurado através de duas chaves hexadecimais S1 e S2 como ilustra a figura 2.2.

- Endereços válidos: 1 a 127 (01h a 7Fh)


**NOTA!**

Caso o endereço seja alterado, ele somente será válido após o produto ser ligado novamente.

O endereço selecionado através das chaves representa um valor hexadecimal. Portanto, o mesmo deve ser convertido para decimal, quando necessário, na parametrização do mestre CANopen.

As chaves que configuram o endereço CANopen do escravo são ilustradas com mais detalhes na figura 2.2, onde a chave S1 é a menos significativa e a chave S2 é a mais significativa:

- Chave S1 = Eh
- Chave S2 = 2h

Como a chave S1 é a menos significativa o endereço formado pelas chaves é 2Eh, ou 46 decimal.



*Figura 2.2: Chaves rotativas para configuração do endereço*

## 2.4 TAXA DE COMUNICAÇÃO

O gerenciador de partidas SCW100 possui uma chave hexadecimal S3 rotativa que permite configurar a taxa de comunicação desejada. De acordo com a posição da chave, esta configuração é programada conforme indicado na tabela 2.2.

**Tabela 2.2:** Configurações da chave S3 para programação da taxa de comunicação

Ajuste da Chave	Taxa de comunicação
0	1 Mbit/s
1	800 Kbit/s
2	500 Kbit/s
3	250 Kbit/s
4	125 Kbit/s
5	100 Kbit/s
6	50 Kbit/s
7	20 Kbit/s


**NOTA!**

Caso esta configuração seja alterada, ela somente será válida após o produto ser ligado novamente.

## 2.5 RESISTOR DE TERMINAÇÃO

O produto possui a chave S4 que pode ser ativada para habilitar o resistor de terminação conforme figura 2.1. A configuração das chaves para habilitar o resistor de terminação é apresentada na tabela 2.3.

*Tabela 2.3: Configurações da chave S4 de habilitação do resistor de terminação*

Ajuste das Chaves	Opção
S4.1 = OFF e S4.2 = OFF	Terminação CAN desligada
S4.1 = ON e S4.2 = ON	Terminação CAN ligada
S4.1 = OFF e S4.2 = ON	Combinação não permitida
S4.1 = ON e S4.2 = OFF	

## 2.6 LEDS DE INDICAÇÃO

O gerenciador de partidas SCW100 tem um LED bicolor (verde e vermelho), mostrado na figura 2.1, as tabelas abaixo mostram o comportamento destes LEDs em função do estado do gerenciador de partidas:

*Tabela 2.4: LED NET - VERDE*

Indicação	Estado	Descrição
Apagado	-	Sem alimentação.
Verde, pisca uma vez	Stopped	Dispositivo está no estado parado. PDOs e SDOs não estão disponíveis neste estado.
Verde, oscilando a cada 200ms	Pre-operational	Dispositivo está no estado Pré-operacional. PDOs não estão disponíveis para comunicação.
Verde, sólido	Operational	Módulo operacional.

*Tabela 2.5: LED NET - VERMELHO*

Indicação	Estado	Descrição
Apagado	Sem erro	Dispositivo está em condições normais de operação.
Vermelho, pisca 1 vez	Warning ou Passive	Indica que a interface CAN está em estado de Warning ou Error Passive. Pode ocorrer, por exemplo, se for o único equipamento ligado à rede CANopen.
Vermelho, pisca 2 vezes	Erro de Node Guarding ou Heartbeat	Controle de erros da comunicação CANopen detectou erro de comunicação utilizando o mecanismo de guarding ou heartbeat.
Vermelho, sólido	Erro de BUS OFF	Indica que a interface CAN está no estado de BUS OFF. Indica uma condição crítica de operação na rede CANopen, em geral associada a problemas na instalação ou configuração incorreta da taxa de comunicação. É necessário desligar e ligar novamente o equipamento para restaurar a comunicação.
Vermelho, oscilando a cada 50ms	CANopen não inicializado	Indica que o Protocolo CANopen não foi inicializado. Verificar se o endereço está ajustado em um valor válido (01h – 7Fh).

### 3 INSTALAÇÃO EM REDE CANOPEN

A rede CANopen, como várias redes de comunicação industriais, pelo fato de ser aplicada muitas vezes em ambientes agressivos e com alta exposição a interferência eletromagnética, exige certos cuidados que devem ser tomados para garantir uma baixa taxa de erros de comunicação durante a sua operação. A seguir são apresentadas recomendações para realizar a instalação do produto na rede.


**NOTA!**

Recomendações detalhadas de como realizar a instalação podem ser obtidas no documento "Planning and Installation Manual" (item DOCUMENTOS).

#### 3.1 TAXA DE COMUNICAÇÃO

Equipamentos com interface CANopen em geral permitem configurar a taxa de comunicação desejada, podendo variar de 10 kbit/s até 1 Mbit/s. A taxa de comunicação (baud rate) que pode ser utilizada por um equipamento também depende do comprimento do cabo utilizado na instalação. A tabela 3.1 apresenta a relação entre as taxas de comunicação e o comprimento máximo de cabo que pode ser utilizado na instalação, de acordo com o recomendado pela especificação do protocolo.

*Tabela 3.1: Taxas de comunicação suportadas e comprimento do cabo*

Taxa de Comunicação	Comprimento do Cabo
10 Kbit/s	1000 m
20 Kbit/s	1000 m
50 Kbit/s	1000 m
100 Kbit/s	600 m
125 kbit/s	500 m
250 kbit/s	250 m
500 kbit/s	100 m
800 Kbit/s	50 m
1 Mbit/s	25 m

Todos os equipamentos da rede devem ser programados para utilizar a mesma taxa de comunicação.

#### 3.2 ENDEREÇO NA REDE CANOPEN

Todo dispositivo na rede CANopen deve possuir um endereço, ou Node-ID, entre 1 e 127. Este endereço precisa ser diferente para cada equipamento.

#### 3.3 RESISTORES DE TERMINAÇÃO

A utilização de resistores de terminação nas extremidades do barramento é fundamental para evitar reflexão de linha, que pode prejudicar o sinal transmitido e ocasionar erros na comunicação. Resistores de terminação no valor de 121  $\Omega$  | 0.25 W devem ser conectados entre os sinais CAN\_H e CAN\_L nas extremidades do barramento principal.

#### 3.4 CABO

Para a ligação dos sinais CAN\_L e CAN\_H deve-se utilizar par trançado com blindagem. A tabela a seguir apresenta as características recomendadas para o cabo.

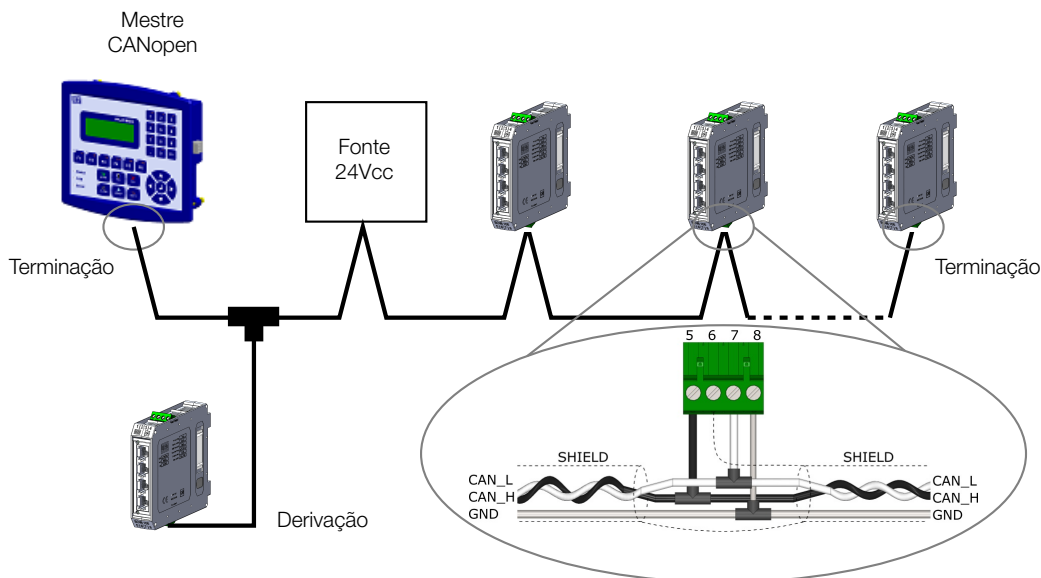
**Tabela 3.2:** Características do cabo para rede CANopen

Comprimento do Cabo (m)	Resistência por Metro (mΩ/m)	Área do Condutor (mm <sup>2</sup> )
0 ... 40	70	0.25 ... 0.34
40 ... 300	<60	0.34 ... 0.60
300 ... 600	<40	0.50 ... 0.60
600 ... 1000	<26	0.75 ... 0.80

Também é necessária a utilização de um par trançado adicional para levar a alimentação de 24Vcc para os equipamentos que necessitam deste sinal. Recomenda-se utilizar um cabo certificado para rede DeviceNet.

### 3.5 LIGAÇÃO NA REDE

Para interligar os diversos nós da rede, recomenda-se a conexão do equipamento diretamente a partir da linha principal, sem a utilização de derivações. Durante a instalação dos cabos, deve-se evitar sua passagem próxima a cabos de potência, pois isto facilita a ocorrência de erros durante a transmissão devido à interferência eletromagnética.


**Figura 3.1:** Exemplo de instalação em rede CANopen

Para evitar problemas de circulação de corrente por diferença de potencial entre diferentes aterramentos, é necessário que todos os dispositivos estejam conectados no mesmo ponto de terra.

Para evitar problemas de diferença de tensão na alimentação entre os dispositivos da rede, é recomendado que a rede seja alimentada em apenas um ponto, e o sinal de alimentação seja levado a todos os dispositivos através do cabo. Caso seja necessária mais de uma fonte de alimentação, estas devem estar referenciadas ao mesmo ponto. É recomendado utilizar uma fonte de alimentação dedicada para alimentação do barramento apenas.

O número máximo de dispositivos conectados em um único segmento da rede é limitado em 64. Repetidores podem ser utilizados para conectar um número maior de dispositivos.

## 4 S STATUS

Permite acessar as variáveis de leitura do gerenciador de partidas SCW100.

### S4 COMUNICAÇÃO

Parâmetros de leitura que apresentam informações sobre as interfaces de comunicação do produto.

#### S4 Comunicação

##### P0030: CAN - Endereço

Faixa de valores: 1 ... 127

Padrão: 1

#### Descrição:

Permite visualizar o endereço utilizado para comunicação CAN do dispositivo, programado através das DIP switches S1 e S2.

#### S4 Comunicação

##### P0031: CAN - Taxa de Comunicação

Faixa de valores: 0 ... 7

Padrão: 0

#### Descrição:

Permite visualizar o valor da taxa de comunicação da interface CAN, programado na DIP switch S3, em bits por segundo.

Indicação	Descrição
0 = 1 Mbit/s	Taxa de comunicação CAN.
1 = 800 Kbit/s	Taxa de comunicação CAN.
2 = 500 Kbit/s	Taxa de comunicação CAN.
3 = 250 Kbit/s	Taxa de comunicação CAN.
4 = 125 Kbit/s	Taxa de comunicação CAN.
5 = 100 Kbit/s	Taxa de comunicação CAN.
6 = 50 Kbit/s	Taxa de comunicação CAN.
7 = 20 Kbit/s	Taxa de comunicação CAN.

#### S4 Comunicação

##### P0033: CAN - Estado do Controlador

Faixa de valores: 0 ... 5

Padrão: 0

#### Descrição:

Permite identificar se a interface CAN está ativa e se a comunicação apresenta erros.

Indicação	Descrição
0 = Inativo	Interface CAN inativa.
1 = Reservado	-
2 = CAN Ativo	Interface CAN ativa e sem erros.
3 = Warning	Controlador CAN atingiu o estado de <i>warning</i> .
4 = Erro Passivo	Controlador CAN atingiu o estado de <i>error passive</i> .
5 = Bus Off	Controlador CAN atingiu o estado de <i>bus off</i> .

#### S4 Comunicação

##### P0034: CAN - Telegramas CAN RX

Faixa de valores: 0 ... 65535

Padrão: 0

#### Descrição:

Este parâmetro funciona como um contador cíclico, que é incrementado toda vez que um telegrama CAN é recebido. Fornece um retorno para o operador se o dispositivo está conseguindo comunicar-se com a rede.

**S4 Comunicação**
**P0035: CAN - Telegramas CAN TX**
**Faixa de valores:** 0 ... 65535

**Padrão:** 0

**Descrição:**

Este parâmetro funciona como um contador cíclico, que é incrementado toda vez que um telegrama CAN é transmitido. Fornece um retorno para o operador se o dispositivo está conseguindo comunicar-se com a rede.

**S4 Comunicação**
**P0036: CAN - Contador Bus Off**
**Faixa de valores:** 0 ... 65535

**Padrão:** 0

**Descrição:**

Contador cíclico que indica o número de vezes que o equipamento entrou em estado de bus off na rede CAN.

**S4 Comunicação**
**P0037: CAN - Telegramas Perdidos**
**Faixa de valores:** 0 ... 65535

**Padrão:** 0

**Descrição:**

Contador cíclico que indica o número de mensagens recebidas pela interface CAN, mas que não puderam ser processadas pelo equipamento. Caso o número de mensagens perdidas seja incrementado com frequência, recomenda-se diminuir a taxa de comunicação utilizada para a rede CAN.

**S4 Comunicação**
**P0038: CAN - Estado da Comunicação CANopen**
**Faixa de valores:** 0 ... 5

**Padrão:** 0

**Descrição:**

Indica o estado do cartão com relação à rede CANopen, informando se o protocolo foi habilitado e se o serviço de controle de erros está ativo (*Node Guarding* ou *Heartbeat*).

Indicação	Descrição
0 = Inativo	Protocolo CANopen desabilitado.
1 = Reservado	-
2 = Comunic. Hab.	Comunicação habilitada.
3 = Ctrl. Erros Hab.	Comunicação habilitada e controle de erros habilitado ( <i>Node Guarding/Heartbeat</i> ).
4 = Erro Guarding	Ocorreu erro de <i>Node Guarding</i> .
5 = Erro Heartbeat	Ocorreu erro de <i>Heartbeat</i> .

**S4 Comunicação**
**P0039: CAN - Estado do Escravo CANopen**
**Faixa de valores:** 0 ... 4

**Padrão:** 0

**Descrição:**

Cada escravo da rede CANopen possui uma máquina de estados que controla o seu comportamento com relação à comunicação. Este parâmetro indica em qual estado encontra-se o dispositivo.

Indicação	Descrição
0 = Inativo	Protocolo CANopen desabilitado.
1 = Inicialização	Não é possível comunicar-se com o dispositivo nesta etapa, que é concluída automaticamente.
2 = Parado	Apenas o objeto NMT está disponível.
3 = Operacional	Todos os objetos de comunicação estão disponíveis.
4 = PreOperacional	É possível comunicar-se com o escravo porém os PDOs ainda não estão disponíveis para operação.

## 5 C CONFIGURAÇÕES

Permite acessar as variáveis de escrita do gerenciador de partidas SCW100.

### C2 COMUNICAÇÃO

Parâmetros de escrita que permitem configurar as interfaces de comunicação do produto.

#### C2 Comunicação

##### **P0101: Ação para Falha na Comunicação**

**Faixa de valores:** 0 ... 1

**Padrão:** 1

#### **Descrição:**

Permite configurar o modo de atuação da proteção para erros de comunicação.

Indicação	Descrição
0 = Sem Ação	Não há atuação.
1 = Desliga Saídas	Desliga todas as saídas do produto. Equivale a receber o comando de parada para as partidas, e zerar a palavra de comando para as saídas digitais.

#### C2 Comunicação

##### **P0032: CAN - Reset de Bus Off**

**Faixa de valores:** 0 ... 1

**Padrão:** 0

#### **Descrição:**

Permite programar qual o comportamento do equipamento ao detectar um erro de *bus off* na interface CAN.

Indicação	Descrição
0 = Manual	Caso ocorra bus off, será indicado esta condição nos LEDs de indicação e a comunicação será desabilitada. A ação programada no parâmetro P0101 - Ação para Falha na Comunicação será executada. Para que o equipamento volte a se comunicar através da interface CAN, será necessário desabilitar e habilitar a interface, ou reiniciar o produto.
1 = Automático	Caso ocorra bus off, a comunicação será reiniciada automaticamente e o erro será ignorado. Neste caso, não haverá indicação nos LEDs e não será executada a ação para erro de comunicação.

## **6 OPERAÇÃO NA REDE CANOPEN**

### **6.1 ACESSO AO DADOS**

Cada escravo da rede CANopen possui uma lista, denominada dicionário de objetos, que contém todos os dados que são acessíveis via rede. Cada objeto desta lista é identificado através de um índice, e durante a configuração do equipamento e troca de mensagens, este índice é utilizado para identificar o que está sendo transmitido.

### **6.2 DADOS CÍCLICOS**

Dados cíclicos são os dados normalmente utilizados para monitoração do estado e controle da operação do equipamento. Para o protocolo CANopen, a interface permite a comunicação de 2 PDOs de recepção e 5 PDOs de transmissão.

É necessário que esta configuração seja feita no mestre da rede CANopen.

### **6.3 DADOS ACÍCLICOS**

Além dos dados cíclicos, a interface também disponibiliza dados acíclicos via SDO. Utilizando este tipo de comunicação, é possível acessar qualquer parâmetro do equipamento. O acesso a este tipo de dado normalmente é feito usando instruções para leitura ou escrita dos dados, onde deve-se indicar o índice e sub-índice para o dado desejado. O item 7.4 descreve como endereçar os parâmetros do gerenciador de partidas SCW100.

### **6.4 OBJETOS RESPONSÁVEIS PELA COMUNICAÇÃO - COB**

Existe um determinado conjunto de objetos que são responsáveis pela comunicação entre os dispositivos da rede. Estes objetos estão divididos de acordo com os tipos de dados e a forma como são enviados ou recebidos por um dispositivo. Os seguintes objetos de comunicação (COBs) são descritos pela especificação:



**Tabela 6.1: Tipos de Objetos de Comunicação (COBs)**

Tipo de Objeto	Descrição
Service Data Object (SDO)	Os SDOs são objetos responsáveis pelo acesso direto ao dicionário de objetos de um dispositivo. Através de mensagens utilizando os SDOs, é possível indicar explicitamente (através do índice do objeto), qual o dado que está sendo manipulado. Existem dois tipos de SDOs: Cliente SDO, responsável por fazer uma requisição leitura ou escrita para um dispositivo da rede, e o Servidor SDO, responsável por atender esta requisição. Como os SDOs são utilizados geralmente para configuração de um nó da rede, são menos prioritários que outros tipos de mensagens.
Process Data Object (PDO)	Os PDOs são utilizados para acessar dados do equipamento sem a necessidade de indicar explicitamente qual o objeto do dicionário está sendo acessado. Para isso, é necessário configurar previamente quais os dados que o PDO estará transmitindo (mapeamento dos dados). Também existem dois tipos de PDOs: PDO de recepção e PDO de transmissão. PDOs usualmente são utilizados para transmissão e recepção de dados utilizados durante a operação do dispositivo, e por isso são mais prioritários que os SDOs.
Emergency Object (EMCY)	Este objeto é responsável pelo envio de mensagens para indicar a ocorrência de erros no dispositivo. Quando um erro ocorre em um determinado dispositivo (Produtor EMCY), este pode enviar uma mensagem para a rede. Caso algum dispositivo da rede esteja monitorando esta mensagem (Consumidor EMCY), é possível programar para que uma ação seja tomada (desabilitar demais dispositivos da rede, reset de erros, etc.).
Synchronization Object (SYNC)	Na rede CANopen é possível programar um dispositivo (Produtor SYNC) para enviar, periodicamente, uma mensagem de sincronização para todos os dispositivos da rede. Estes dispositivos (Consumidores SYNC) podem então, por exemplo, enviar um determinado dado que necessita ser disponibilizado periodicamente.
Network Management (NMT)	Toda a rede CANopen precisa ter um mestre que controle os demais dispositivos da rede (escravos). Este mestre será responsável por um conjunto de serviços que controlam a comunicação dos escravos e seu estado na rede CANopen. Os escravos são responsáveis por receber os comandos enviados pelo mestre e executar as ações solicitadas. Dentre os serviços descritos pelo protocolo estão: serviços de controle do dispositivo, onde o mestre controla o estado de cada escravo na rede, e serviços de controle de erros (Node Guarding e Heartbeat), onde o dispositivo envia mensagens periódicas para informar que a conexão está ativa.

Toda a comunicação do escravo com a rede é feita utilizando estes objetos, e os dados que podem ser acessados são os existentes no dicionário de objetos do dispositivo.

## 6.5 COB-ID

Um telegrama da rede CANopen sempre é transmitido por um objeto de comunicação (COB). Todo COB possui um identificador que indica o tipo de dado que está sendo transportado. Este identificador, chamado de COB-ID, possui um tamanho de 11 bits, e é transmitido no campo identificador de um telegrama CAN. Ele pode ser subdividido em duas partes:

Código da Função				Endereço do nó						
bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- Código da função: indica o tipo de objeto que está sendo transmitido.
- Endereço do nó: indica com qual dispositivo da rede o telegrama está vinculado.

A seguir é apresentada uma tabela com os valores padrão para os diferentes objetos de comunicação. É necessário observar que o valor padrão do objeto depende do endereço do escravo, com exceção dos COB-IDs para NMT e SYNC, que são comuns para todos os elementos da rede. Estes valores também podem ser alterados durante a etapa de configuração do dispositivo.

**Tabela 6.2:** COB-ID para os diferentes objetos

<b>COB</b>	<b>Código da Função (bits 10-7)</b>	<b>COB-ID Resultante (função + endereço)</b>
NMT	0000	0
SYNC	0001	128 (80h)
EMCY	0001	129 - 255 (81h - FFh)
PDO1 (tx)	0011	385 - 511 (181h - 1FFh)
PDO1 (rx)	0100	513 - 639 (201h - 27Fh)
PDO2 (tx)	0101	641 - 767 (281h - 2FFh)
PDO2 (rx)	0110	769 - 895 (301h - 37Fh)
PDO3 (tx)	0111	897 - 1023 (381h - 3FFh)
PDO3 (rx)	1000	1025 - 1151 (401h - 47Fh)
PDO4 (tx)	1001	1153 - 1279 (481h - 4FFh)
PDO4 (rx)	1010	1281 - 1407 (501h - 57Fh)
SDO (tx)	1011	1409 - 1535 (581h - 5FFh)
SDO (rx)	1100	1537 - 1663 (601h - 67Fh)
Node Guarding/Heartbeat	1110	1793 - 1919 (701h - 77Fh)

## 6.6 ARQUIVO EDS

Cada dispositivo em uma rede CANopen possui um arquivo de configuração EDS, que contém informações sobre o funcionamento do dispositivo na rede. Em geral este arquivo é utilizado por um mestre ou software de configuração, para programação dos dispositivos presentes na rede CANopen.

O arquivo de configuração EDS está disponível na página de internet da WEG (<http://www.weg.net>). É importante observar se o arquivo de configuração EDS é compatível com a versão de firmware do gerenciador de partidas SCW100.

## 7 DICIONÁRIO DE OBJETO

O dicionário de objetos é uma lista com os diversos dados do equipamento que são acessíveis através da rede CANopen. Um objeto desta lista é identificado através de um índice de 16 bits, e é baseado nesta lista que toda a troca de dados entre os dispositivos é efetuada.

O documento CiA DS 301 define um conjunto mínimo de objetos que todo o escravo da rede CANopen deve possuir. Os objetos disponíveis nesta lista são agrupados de acordo com o tipo de função que ele executa. Os objetos são dispostos no dicionário da seguinte maneira:

*Tabela 7.1: Agrupamento do dicionário de objetos*

Índice	Objetos	Descrição
0001h - 025Fh	Definição dos tipos de dados	Utilizado como referência para os tipos de dados suportados pelo sistema.
1000h - 1FFFh	Objetos de comunicação	São objetos comuns a todos os dispositivos CANopen. Contém informações gerais sobre o equipamento e também dados para a configuração da comunicação.
2000h - 5FFFh	Objetos específicos do fabricante	Nesta faixa, cada fabricante de equipamentos CANopen é livre para definir quais dados estes objetos representarão.
6000h - 9FFFh	Objetos padronizados para dispositivos	Esta faixa é reservada para objetos que descrevem o comportamento de equipamentos similares, independente do fabricante.

Demais índices não referenciados nesta lista são reservados.

### 7.1 ESTRUTURA DO DICIONÁRIO

A estrutura geral do dicionário de objetos possui o seguinte formato:

Índice	Objeto	Nome	Tipo	Acesso
--------	--------	------	------	--------

- **Índice:** índice diretamente o índice do objeto no dicionário.
- **Objeto:** descreve que informação o índice armazena (variável simples, array, record, etc.).
- **Nome:** contém o nome do objeto para facilitar sua identificação.
- **Tipo:** indica diretamente o tipo de dado armazenado. Para variáveis simples, este tipo pode ser um inteiro, um float, etc. Para arrays, ele indica o tipo do dado contido no array. Para records, ele indica o formato do record, de acordo com os tipos descritos na primeira parte do dicionário de objetos (índices 0001h – 025Fh).
- **Acesso:** informa se o objeto em questão está acessível somente para leitura (ro), para leitura e escrita (rw), ou é uma constante (const).

Para objetos do tipo array ou records, ainda é necessário um sub-índice, que não é descrito na estrutura do dicionário.

### 7.2 TIPOS DE DADOS

A primeira parte do dicionário de objetos (índices 0001h – 025Fh) descreve os tipos de dados que podem ser acessados em um dispositivo na rede CANopen. Estes podem ser tipos básicos, como inteiros e floats, ou tipos compostos, formados por um conjunto de entradas, como records e arrays.

### 7.3 COMMUNICATION PROFILE - OBJETOS PARA COMUNICAÇÃO

Os índices de 1000h até 1FFFh correspondem, no dicionário de objetos, à parte responsável pelas configurações da comunicação na rede CANopen. Estes objetos são comuns a todos os dispositivos, mas somente alguns são

obrigatórios. A seguir é apresentada uma lista com alguns dos objetos desta faixa suportados pelo gerenciador de partidas SCW100.

**Tabela 7.2:** Lista de objetos – Communication Profile

Índice	Objeto	Nome	Tipo	Acesso
1000h	VAR	device type	UNSIGNED32	ro
1001h	VAR	error register	UNSIGNED8	ro
1005h	VAR	COB-ID SYNC	UNSIGNED32	rw
100Ch	VAR	guard time	UNSIGNED16	rw
100Dh	VAR	life time factor	UNSIGNED8	rw
1016h	ARRAY	consume heartbeat time	UNSIGNED32	rw
1017h	VAR	producer heartbeat time	UNSIGNED16	rw
1018h	RECORD	Identity Object	Identity	ro
Server SDO Parameter				
1200h	RECORD	1st Server SDO parameter	SDO Parameter	ro
Receive PDO Communication Parameter				
1400h	RECORD	1st receive PDO Parameter	PDO CommPar	rw
1401h	RECORD	2nd receive PDO Parameter	PDO CommPar	rw
Receive PDO Mapping Parameter				
1600h	RECORD	1st receive PDO mapping	PDO Mapping	rw
1601h	RECORD	2nd receive PDO mapping	PDO Mapping	rw
Transmit PDO Communication Parameter				
1800h	RECORD	1st transmit PDO Parameter	PDO CommPar	rw
1801h	RECORD	2nd transmit PDO Parameter	PDO CommPar	rw
1802h	RECORD	3rd transmit PDO Parameter	PDO CommPar	rw
1803h	RECORD	4th transmit PDO Parameter	PDO CommPar	rw
1804h	RECORD	5th transmit PDO Parameter	PDO CommPar	rw
Transmit PDO Mapping Parameter				
1A00h	RECORD	1st transmit PDO mapping	PDO Mapping	rw
1A01h	RECORD	2nd transmit PDO mapping	PDO Mapping	rw
1A02h	RECORD	3rd transmit PDO mapping	PDO Mapping	rw
1A03h	RECORD	4th transmit PDO mapping	PDO Mapping	rw
1A04h	RECORD	5th transmit PDO mapping	PDO Mapping	rw

Estes objetos somente podem ser lidos e escritos através da rede CANopen, não estão disponíveis em outra interface de rede. O mestre da rede, em geral, é o equipamento responsável pela configuração do equipamento antes de iniciar a operação. O arquivo de configuração EDS traz a lista de todos os objetos de comunicação suportados.

Para uma descrição detalhada de quais objetos estão disponíveis nesta faixa do dicionário de objetos, consulte o item 8.

## 7.4 OBJETOS ESPECÍFICOS DO FABRICANTE

Nos índices de 2000h até 5FFFh, cada fabricante é livre para definir quais objetos estarão presentes, o tipo e a função de cada objeto. Para o gerenciador de partidas SCW100, nesta faixa de objetos foi disponibilizada toda a lista de parâmetros. Através destes parâmetros é possível operar o equipamento, executando qualquer função que o SCW100 possa realizar. Os parâmetros foram disponibilizados a partir do índice 2000h, e somando o Net Id a este índice se obtém sua posição no dicionário. Para identificar como estão distribuídos os parâmetros no dicionário de objetos consultar o item A.

É necessário conhecer a operação do SCW100 através dos parâmetros para poder programar corretamente sua operação via rede CANopen.

Para a lista completa e uma descrição detalhada dos parâmetros, consulte o manual de programação do gerenciador de partidas SCW100.

## 8 DESCRIÇÃO DOS OBJETOS DE COMUNICAÇÃO

Neste item são descritos detalhadamente cada um dos objetos de comunicação disponíveis para o gerenciador de partidas SCW100. É necessário conhecer como estes objetos são operados para utilizar as funções disponíveis para a comunicação do SCW100.

### 8.1 OBJETOS DE IDENTIFICAÇÃO

Existe um conjunto de objetos no dicionário utilizados para identificação do equipamento, porém não possuem influência no seu comportamento na rede CANopen.

#### 8.1.1 Objeto 1000h - Device Type

Este objeto fornece um código em 32 bits que descreve o tipo de objeto e sua funcionalidade.

*Tabela 8.1: Objeto 1000h - Device Type*

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1000h	0	Device Type	UNSIGNED32	RO	No	0

Este código pode ser dividido em duas partes: 16 bits inferiores, descrevendo o tipo de perfil (profile) que o dispositivo utiliza, e 16 bits superiores, indicando uma função específica, de acordo com o perfil especificado.

#### 8.1.2 Objeto 1001h - Error Register

Este objeto indica a ocorrência ou não de erro no dispositivo. O tipo de erro registrado para o equipamento segue o descrito pela tabela 8.2.

*Tabela 8.2: Objeto 1001h - Error Register*

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1001h	0	Error register	UNSIGNED8	RO	yes	0

*Tabela 8.3: Estrutura do objeto Error Register*

Bit	Significado
0	Erro genérico
1	Corrente
2	Tensão
3	Temperatura
4	Comunicação
5	Reservado (sempre 0)
6	Reservado (sempre 0)
7	específico do fabricante

Caso o dispositivo apresente algum erro, o bit equivalente deve ser ativado. O primeiro bit (erro genérico) deverá ser ativado em qualquer situação de erro.

### 8.1.3 Objeto 1018h - Identity Object

Traz informações gerais sobre o dispositivo.

*Tabela 8.4: Objeto 1018h - Identity Object*

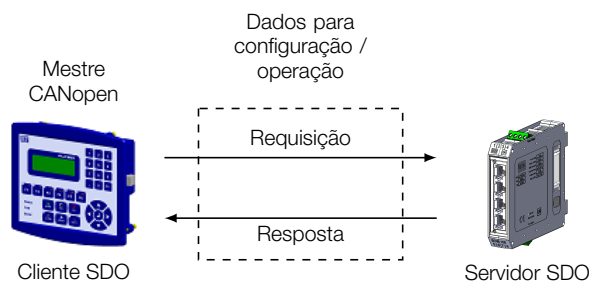
Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1018h	0	Número do último sub-índice	UNSIGNED8	RO	No	4
	1	Vendor ID	UNSIGNED32	RO	No	0000.0123h
	2	Código do produto	UNSIGNED32	RO	No	0000.1200h
	3	Número da revisão	UNSIGNED32	RO	No	De acordo com a versão de firmware do equipamento
	4	Número serial	UNSIGNED32	RO	No	Diferente para cada gerenciador de partidas SCW100 em rede CANopen

O Vendor ID é um número que identifica o fabricante junto à CiA. O código do produto é definido pelo fabricante de acordo com o tipo de produto. O número da revisão representa a versão de firmware do equipamento. O sub-índice 4 é um número serial único para cada gerenciador de partidas SCW100 em rede CANopen.

## 8.2 SERVICE DATA OBJECTS - SDOS

Os SDOs são responsáveis pelo acesso direto ao dicionário de objetos de um determinado dispositivo na rede. Eles são utilizados para a configuração e, portanto, possuem baixa prioridade, já que não devem ser utilizados para comunicar dados necessários para a operação do dispositivo.

Existem dois tipos de SDOs: cliente e servidor. Basicamente, a comunicação inicia com o cliente (usualmente o mestre da rede) fazendo uma requisição de leitura (upload) ou escrita (download) para um servidor, e este responde ao que foi requisitado.



*Figura 8.1: Comunicação entre cliente e servidor SDO*

### 8.2.1 Objeto 1200h - Servidor SDO

O gerenciador de partidas SCW100 possui um único SDO do tipo servidor, que possibilita o acesso a todo o seu dicionário de objetos. Através dele, um cliente SDO pode configurar a comunicação, parâmetros e modos de operação do SCW100. Todo servidor SDO possui um objeto, do tipo SDO\_PARAMETER, para a sua configuração, possuindo a seguinte estrutura:

*Tabela 8.5: Objeto 1200h - Servidor SDO*

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1200h	0	Número do último sub-índice	UNSIGNED8	RO	No	2
	1	COB-ID Cliente - Servidor (rx)	UNSIGNED32	RO	No	600h + Node-ID
	2	COB-ID Servidor - Cliente (tx)	UNSIGNED32	RO	No	580h + Node-ID

### 8.2.2 Funcionamento dos SDOs

Um telegrama enviado por um SDO possui 8 bytes de tamanho, com a seguinte estrutura:

Identificador	8 bytes de dados							
11 bits	Comando	Índice		Sub-índice	Dados do objeto			
	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

O identificador depende do sentido da transmissão (rx ou tx) e do endereço (ou Node-ID) do servidor destino. Por exemplo, um cliente que faz uma requisição para um servidor cujo Node-ID é 1, deve enviar uma mensagem com o identificador igual a 601h. O servidor irá receber esta mensagem e responder com um telegrama cujo COB-ID é igual a 581h.

O código do comando depende do tipo de função utilizada. Para as transmissões de um cliente para um servidor, podem ser utilizados os seguintes comandos:

**Tabela 8.6:** Código dos comandos para cliente SDO

Comando	Função	Descrição	Dados do Objeto
22h	Download	Escrita em objeto	Indefenido
23h	Download	Escrita em objeto	4 byte
2Bh	Download	Escrita em objeto	2 byte
2Fh	Download	Escrita em objeto	1 byte
40h	Upload	Leitura de objeto	Não utilizado
60h ou 70h	Upload segment	Leitura segmentada	Não utilizado

Ao fazer a requisição, o cliente indicará através de seu COB-ID, qual o endereço do escravo para o qual esta requisição se destina. Somente um escravo (usando seu respectivo servidor SDO) poderá responder para o cliente o telegrama recebido. O telegrama de resposta possuirá também a mesma estrutura do telegrama de requisição, mas os comandos serão diferentes:

**Tabela 8.7:** Código dos comandos para servidor SDO

Comando	Função	Descrição	Dados do Objeto
60h	Download	Resposta para escrita em objeto	Não utilizado
43h	Upload	Resposta para escrita em objeto	4 byte
4Bh	Upload	Resposta para escrita em objeto	2 byte
4Fh	Upload	Resposta para escrita em objeto	1 byte
41h	Upload segment	Inicia resposta segmentada para leitura	4 byte
01h ou 0Dh	Upload segment	Último segmento de dados para leitura	8 ... 2 bytes

Para leituras que envolvem até quatro bytes de dados, uma única mensagem pode ser transmitida pelo servidor; para leitura de uma quantidade maior de bytes, é necessário que cliente e servidor troquem múltiplos telegramas.

Um telegrama somente é completo após a confirmação do servidor para a requisição feita pelo cliente. Caso algum erro seja detectado durante a troca de telegramas (por exemplo, não há resposta do servidor), o cliente poderá abortar o processo com uma mensagem de aviso com o código do comando igual a 80h.



**NOTA!**

Quando o SDO é utilizado para escrita nos objetos que representam os parâmetros do SCW100 (objetos a partir do índice 2000h), este valor é salvo na memória não volátil do produto. Desta forma, depois de desligado ou feito o reset do equipamento, os valores configurados não são perdidos. Para os demais objetos, estes valores não são salvos automaticamente, de maneira que é necessário reescrever os valores desejados.

Exemplo: um cliente SDO solicita para um escravo no endereço 1 a leitura do objeto identificado pelo índice 2000h,

sub-índice 0 (zero), que representa um inteiro de 16 bits. O telegrama do mestre possui a seguinte forma:

Identificador	Comando	Índice		Sub-índice	Dados			
601h	40h	00h	20h	00h	00h	00h	00h	00h

O escravo responde à requisição, indicando que o valor para o referido objeto é igual a 999<sup>1</sup>:

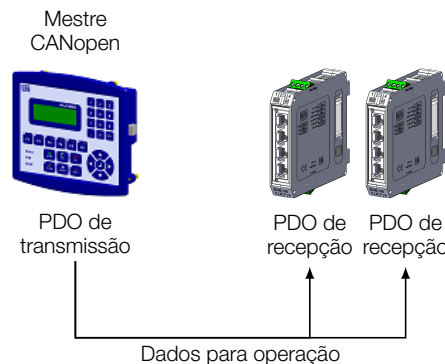
Identificador	Comando	Índice		Sub-índice	Dados			
581h	4Bh	00h	20h	00h	E7h	03h	00h	00h

### 8.3 PROCESS DATA OBJECTS - PDOS

Os PDOS são utilizados para enviar e receber dados utilizados durante a operação do dispositivo, que muitas vezes precisam ser transmitidos de forma rápida e eficiente. Por isso, eles possuem uma prioridade maior do que os SDOs.

Nos PDOS, apenas os dados são transmitidos no telegrama (índices e sub-índices são omitidos), e desta forma é possível fazer uma transmissão mais eficiente, com maior volume de dados em um único telegrama. É necessário, porém, configurar previamente o que está sendo transmitido pelo PDO, de forma que, mesmo sem a indicação do índice e sub-índice, seja possível saber o conteúdo do telegrama.

Existem dois tipos de PDOS, os PDOS de recepção e os PDOS de transmissão. Os PDOS de transmissão são responsáveis por enviar dados para a rede, enquanto que os PDOS de recepção ficam responsáveis por receber e tratar estes dados. Desta forma é possível que haja comunicação entre escravos da rede CANopen, basta configurar um escravo para transmitir uma informação, e um ou mais escravos para receber esta informação.



**Figura 8.2:** Comunicação utilizando PDOS



**NOTA!**

PDOS somente podem ser transmitidos ou recebidos quando o dispositivo está no estado operacional.

#### 8.3.1 Objetos Mapeáveis para os PDOS

Para um objeto poder ser transmitido através de um PDO, é necessário que ele seja mapeável para o conteúdo do PDO. Na descrição dos objetos de comunicação (1000h – 1FFFh), o campo “Mapeável” informa esta condição. Usualmente, apenas informações necessárias para a operação do dispositivo são mapeáveis, como comandos para habilitação, status do dispositivo, referências, etc. Informações para configuração do dispositivo não são acessíveis através de PDOS, e caso seja necessário acessá-las via rede deve-se utilizar os SDOs.

<sup>1</sup> Não esquecer que qualquer dado do tipo inteiro, a ordem de transferência dos bytes vai do menos significativo até o mais significativo.



Para os objetos específicos do fabricante (2000h – 5FFFh), a tabela A apresenta os objetos mapeáveis para os PDOs. Parâmetros com acesso apenas para leitura (ro) podem ser utilizados apenas por PDOs de transmissão, enquanto que os demais parâmetros podem ser utilizados apenas por PDOs de recepção.

O arquivo EDS do equipamento traz a lista de todos os objetos disponíveis, informando se o objeto é mapeável ou não.

### 8.3.2 PDOs de Recepção

Os PDOs de recepção, ou RPDOs, são responsáveis por receber dados que outros dispositivos enviam para a rede CANopen. O gerenciador de partidas SCW100 possui 2 PDOs de recepção, cada um podendo receber até 8 bytes de dados. Cada RPDO possui dois parâmetros para sua configuração, um PDO\_COMM\_PARAMETER e um PDO\_MAPPING, conforme descrito a seguir.

PDO\_COMM\_PARAMETER

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1400h - 1401h	0	Número do último sub-índice	UNSIGNED8	RO	No	2
	1	COB-ID usado pelo PDO	UNSIGNED32	RW	No	200h / 300h + Node-ID
	2	Tipo de transmissão	UNSIGNED8	RW	No	254

O sub-índice 1 contém o COB-ID do PDO de recepção. Sempre que uma mensagem for enviada para a rede, este objeto irá ler qual o COB-ID desta mensagem, e caso ele seja igual ao valor deste campo, a mensagem será recebida pelo dispositivo. Este campo é formado por um UNSIGNED32 com a seguinte estrutura:

**Tabela 8.8:** Descrição do COB-ID

Bit	Valor	Descrição
31 (MSB)	0	PDO está habilitado
	1	PDO está desabilitado
30	0	RTR permitido
29	0	Tamanho do identificador = 11 bits
28 - 11	0	Não utilizado, sempre 0
10 - 0 (LSB)	X	COB-ID de 11 bits

O bit 31 permite habilitar ou desabilitar o PDO. Os bits 30 e 29, que devem ser mantidos em 0 (zero), indicam respectivamente que o PDO aceita frames remotos (RTR frames) e que utiliza identificador de 11 bits. Como o SCW100 não utiliza identificadores de 29 bits, os bits de 28 até 11 devem ser mantidos em 0 (zero), enquanto que os bits de 10 até 0 (zero) são usados para configurar o COB-ID para o PDO.

O sub-índice 2 indica o tipo de transmissão deste objeto, de acordo com a tabela a seguir.

**Tabela 8.9:** Descrição do tipo de transmissão

Tipo de transmissão	Transmissão de PDOs				
	Cíclico	Acíclico	Síncrono	Assíncrono	RTR
0		•	•		
1 - 240	•		•		
241 - 251	Reservado				
252			•		•
253				•	•
254				•	
255				•	

- **Valores 0 – 240:** qualquer RPDOs programado nesta faixa possui o mesmo funcionamento. Ao detectar uma

mensagem, ele irá receber os dados, porém não atualizará os valores recebidos até detectar o próximo telegrama SYNC.

- **Valores 252 e 253:** não permitido para PDOs de recepção.
- **Valores 254 e 255:** indica que não possui relação com o objeto de sincronização. Ao receber uma mensagem, seus valores serão atualizados imediatamente.

#### PDO\_MAPPING

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1600h - 1601h	0	Número de objetos mapeados	0 = desabilitado 1-4=número de objetos mapeados	RO	No	0
	1 - 4	1º até 4º objeto mapeado no PDO	UNSIGNED32	RW	No	Indicado no arquivo EDS

Este parâmetro indica os objetos mapeados nos PDOs de recepção o gerenciador de partidas SCW100. O valor padrão desses objetos é indicado no arquivo EDS do produto. Para cada RPDO, é possível mapear até 4 objetos diferentes, desde que o tamanho total não ultrapasse oito bytes. O mapeamento de um objeto é feito indicando o seu índice, sub-índice<sup>2</sup> e tamanho (em bits) em um campo UNSIGNED32, com o seguinte formato:

UNSIGNED32		
Índice (16 bits)	Sub-índice (8 bits)	tamanho do objeto (8 bits)

Por exemplo, supondo um mapeamento para um PDO de recepção com os seguintes valores configurados, temos:

- **Sub-índice 0 = 3:** o RPDO possui três objetos mapeados.
- **Sub-índice 1 = 244C.0010h:** o primeiro objeto mapeado possui índice igual a 244Ch, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P1100 - Comando de Partida Direta.
- **Sub-índice 2 = 2451.0010h:** o segundo objeto mapeado possui índice igual a 2451h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P1105 - Comando de Partida Reversa.
- **Sub-índice 3 = 2456.0010h:** o segundo objeto mapeado possui índice igual a 2456h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P1110 - Comando de Parada.

É possível modificar este mapeamento, alterando a quantidade ou o número dos objetos mapeados. Lembrar que no máximo podem ser mapeados 4 objetos ou 8 bytes.



#### NOTA!

- Para poder alterar os objetos mapeados em um PDO, primeiro é necessário escrever o valor 0 (zero) no sub-índice 0 (zero). Desta forma, os valores dos sub-índices 1 até 4 podem ser alterados. Depois de feito o mapeamento desejado, deve-se escrever novamente no sub-índice 0 (zero) o número de objetos que foram mapeados, habilitando novamente o PDO.
- Não esquecer que os PDOs somente podem ser recebidos caso o dispositivo esteja no estado operacional.

### 8.3.3 PDOs de Transmissão

Os PDOs de transmissão, ou TPDOs, como o nome diz, são responsáveis por transmitir dados para a rede CANopen. O gerenciador de partidas SCW100 possui 5 PDOs de transmissão, cada um podendo transmitir até 8 bytes

<sup>2</sup>Caso o objeto seja do tipo VAR e não possua sub-índice, deve ser indicado o valor 0 (zero) para o sub-índice.

de dados. De forma semelhante aos RPDOs, cada TPDO possui dois parâmetros para sua configuração, um PDO\_COMM\_PARAMETER e um PDO\_MAPPING, conforme descrito a seguir.

#### PDO\_COMM\_PARAMETER

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1800h - 1804h	0	Número do último sub-índice	UNSIGNED8	RO	No	5
	1	COB-ID usado pelo PDO	UNSIGNED32	RW	No	180h / 280h / 380h / 480h / 580h + Node-ID
	2	Tipo de transmissão	UNSIGNED8	RW	No	254
	3	Tempo entre transmissões	UNSIGNED16	RW	No	-
	4	Entrada de compatibilidade	UNSIGNED8	RW	No	-
	5	Temporizador de eventos	UNSIGNED16	RW	No	0

O sub-índice 1 contém o COB-ID do PDO de transmissão. Sempre que este PDO enviar uma mensagem para a rede, o identificador desta mensagem será este COB-ID. A estrutura deste campo é descrita na tabela 8.8.

O sub-índice 2 indica o tipo de transmissão deste objeto, que segue o descrito pela tabela 8.9. Porém seu funcionamento é diferente para PDOs de transmissão:

- **Valor 0:** indica que a transmissão deve ocorrer imediatamente após a recepção de um telegrama SYNC, mas não periodicamente.
- **Valores 1 – 240:** o PDO deve ser transmitido a cada telegrama SYNC detectado (ou ocorrências múltiplas de SYNC, de acordo com o número escolhido entre 1 e 240).
- **Valor 252:** indica que o conteúdo da mensagem deve ser atualizado (mas não enviado), após a recepção de um telegrama SYNC. O envio da mensagem deve ser feito após a recepção de um frame remoto (RTR frame).
- **Valor 253:** o PDO deve atualizar e enviar uma mensagem assim que receber um frame remoto.
- **Valor 254:** o objeto deve ser transmitido de acordo com o timer programado no sub-índice 5.
- **Valor 255:** o objeto é transmitido automaticamente quando o valor de algum dos objetos mapeados neste PDO for alterado. Funciona por alteração de estado (Change Of State). Este tipo também permite que o PDO seja transmitido de acordo com o timer programado no sub-índice 5.

No sub-índice 3 é possível programar um tempo mínimo (em múltiplos de 100 µs) que deve transcorrer para que, depois de transmitido um telegrama, um novo telegrama possa ser enviado por este PDO. O valor 0 (zero) desabilita esta função.

O sub-índice 4 não tem função e existe apenas por razões de compatibilidade.

O sub-índice 5 contém um valor para habilitar um temporizador para o envio automático de um PDO. Desta forma, sempre que um PDO for configurado para o tipo assíncrono, é possível programar o valor deste temporizador (em múltiplos de 1 ms), para que o PDO seja transmitido periodicamente no tempo programado.



#### NOTA!

- Deve-se observar o tempo programado neste temporizador, de acordo com a taxa de transmissão utilizada. Tempos muito pequenos (próximos ao tempo de transmissão do telegrama) podem monopolizar o barramento, causando a retransmissão indefinida do PDO e impedindo que outros objetos menos prioritários possam transmitir seus dados.
- O tempo mínimo permitido para esta função no gerenciador de partidas SCW100 é 2 ms.
- É importante observar o tempo entre transmissões programado no sub-índice 3 principalmente quando o PDO for programado com o valor 255 no sub-índice 2 (Change Of State).
- Não esquecer que os PDOs somente podem ser transmitidos caso o escravo esteja no estado operacional.

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1A00h - 1A04h	0	Número do último sub-índice	UNSIGNED8	RO	No	0
	1 - 4	1º até 4º objeto mapeado no PDO	UNSIGNED32	RW	No	0

O PDO MAPPING para a transmissão funciona de forma semelhante ao de recepção, porém neste caso são definidos os dados a serem transmitidos pelo PDO. Cada objeto mapeado deve ser colocado na lista de acordo com o descrito a seguir:

UNSIGNED32		
Índice (16 bits)	Sub-índice (8 bits)	tamanho do objeto (8 bits)

Por exemplo, supondo um mapeamento para um PDO de transmissão com os seguintes valores configurados, temos:

- **Sub-índice 0 = 4:** o TPDO possui quatro objetos mapeados.
- **Sub-índice 1 = 20C8.0010h:** o primeiro objeto mapeado possui índice igual a 20C8h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0200 - Status P1.
- **Sub-índice 2 = 20C9.0010h:** o segundo objeto mapeado possui índice igual a 20C9h, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0201 - Status P1 - Contator.
- **Sub-índice 3 = 20CA.0010h:** o segundo objeto mapeado possui índice igual a 20CAh, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0202 - Status P1 - Erro.
- **Sub-índice 4 = 20CB.0010h:** o segundo objeto mapeado possui índice igual a 20CBh, sub-índice 0 (zero), e tamanho igual a 16 bits. Este objeto corresponde ao parâmetro P0203 - Status P1 - Alarme.

É possível modificar este mapeamento, alterando a quantidade ou o número dos objetos mapeados. Lembrar que no máximo podem ser mapeados 4 objetos ou 8 bytes.

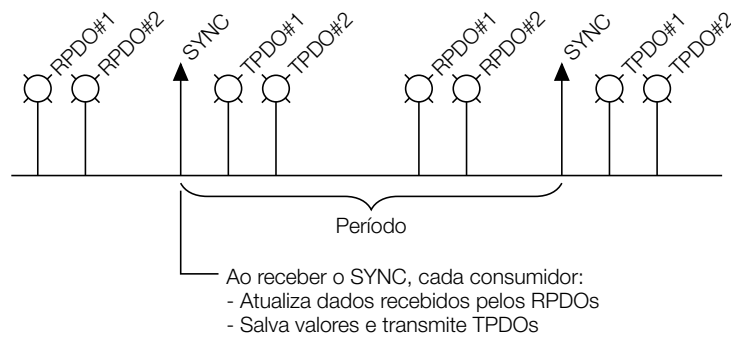

**NOTA!**

Para poder alterar os objetos mapeados em um PDO, primeiro é necessário escrever o valor 0 (zero) no sub-índice 0 (zero). Desta forma, os valores dos sub-índices 1 até 4 podem ser alterados. Depois de feito o mapeamento desejado, deve-se escrever novamente no sub-índice 0 (zero) o número de objetos que foram mapeados, habilitando novamente o PDO.

## 8.4 SYNCHRONIZATION OBJECT - SYNC

Este objeto é transmitido com o objetivo de permitir a sincronização de eventos entre os dispositivos da rede CANopen. Ele é transmitido por um produtor SYNC, e os dispositivos que detectam a sua transmissão são denominados consumidores SYNC.

O gerenciador de partidas SCW100 possui a função de consumidor SYNC e, portanto, pode programar seus PDOs para serem síncronos. PDOs síncronos são aqueles relacionados com o objeto de sincronização e, portanto, podem ser programados para serem transmitidos ou atualizados com base neste objeto.



**Figura 8.3:** SYNC

A mensagem SYNC transmitida pelo produtor não possui dado algum em seu campo de dados, pois seu objetivo é fornecer um evento sincronizado entre os dispositivos da rede. O seguinte objeto está disponível para configuração do consumidor SYNC:

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1005h	0	COB-ID SYNC	UNSIGNED32	RW	No	80h



**NOTA!**

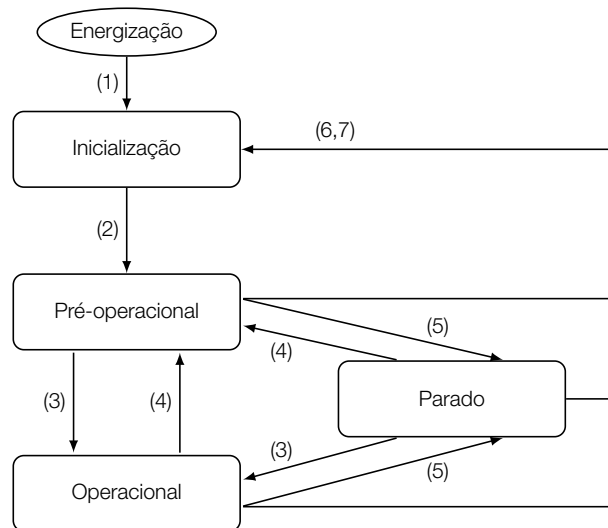
Deve-se observar o tempo programado no produtor para o período dos telegramas SYNC, de acordo com a taxa de transmissão utilizada e o número de PDOs síncronos a serem transmitidos. É necessário que haja tempo suficiente para a transmissão destes objetos, e também é recomendado que haja folga para possibilitar o envio de mensagens assíncronas, como EMCY, PDOs assíncronos e SDOs.

## 8.5 NETWORK MANAGEMENT - NMT

O objeto de gerenciamento da rede é responsável por um conjunto de serviços que controlam a comunicação do dispositivo na rede CANopen. Para este objeto estão disponíveis os serviços de controle do nó e de controle de erros (utilizando Node Guarding ou Heartbeat).

### 8.5.1 Controle dos Estados do Escravo

Com relação à comunicação, um dispositivo da rede CANopen pode ser descrito pela seguinte máquina de estados:



**Figura 8.4:** Diagrama de estados do nó CANopen

**Tabela 8.10:** Descrição das transições

Transição	Descrição
1	Dispositivo é ligado e começa a inicialização (automático)
2	Inicialização concluída, vai para o estado pré-operacional (automático)
3	Recebe comando Start Node para entrar no estado operacional
4	Recebe comando Enter Pre-Operational, e vai para o estado pré-operacional
5	Recebe comando Stop Node para entrar no estado parado
6	Recebe comando Reset Node, onde executa o reset completo do dispositivo
7	Recebe comando Reset Communication, onde reinicializa o valor dos objetos e a comunicação CANopen do dispositivo

Durante a inicialização, é definido o Node-ID, criados os objetos e configurada a interface com a rede CAN. Não é possível comunicar-se com o dispositivo nesta etapa, que é concluída automaticamente. No final desta etapa, o escravo envia para rede um telegrama do objeto Boot-up, utilizado apenas para indicar que a inicialização foi concluída e que o escravo entrou no estado pré-operacional. Este telegrama possui identificador 700h + Node-ID, e apenas um byte de dados com valor igual a 0 (zero).

No estado pré-operacional, já é possível comunicar-se com o escravo. Os PDOs, porém, ainda não estão disponíveis para operação. No estado operacional, todos os objetos estão disponíveis, enquanto que no estado parado, apenas o objeto NMT pode receber ou transmitir telegramas para a rede. A tabela a seguir mostra os objetos disponíveis para cada estado.

**Tabela 8.11:** Objetos acessíveis em cada estado

	Inicialização	Pré-operação	Operacional	Parado
PDO			•	
SDO		•	•	
SYNC		•	•	
EMCY		•	•	
Boot-up	•			
NMT		•	•	•

Esta máquina de estados é controlada pelo mestre da rede, que envia, para cada escravo, comandos para que seja executada a transição de estados desejada. Estes telegramas não possuem confirmação, o que significa que o escravo apenas recebe o telegrama sem retornar resposta para o mestre. Os telegramas recebidos possuem a seguinte estrutura:

Identificador	byte 1	byte 2
00h	Código do comando	Node-ID destino

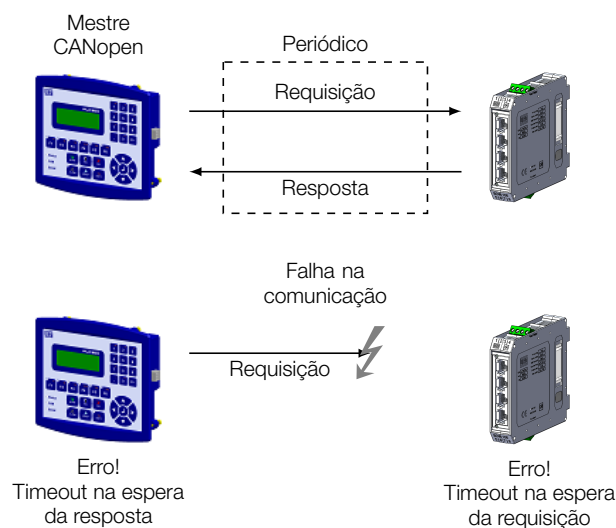
**Tabela 8.12:** Comandos para a transição de estados

Código do comando	Node ID destino
1 = START node (transição 3)	0 = Todos os escravos
2 = STOP node (transição 4)	1 ... 127 = escravo específico
128 = Enter pre-operacional (transição 5)	
129 = Reset node (transição 6)	
130 = Reset communication (transição 7)	

As transições indicadas no código do comando equivalem às transições de estado executadas pelo nó após receber o comando (conforme figura 8.4). O comando Reset node faz com que o escravo execute um reset completo do dispositivo, enquanto que o comando Reset communication faz com que o escravo reinicialize apenas os objetos relativos à comunicação CANopen.

### 8.5.2 Controle de Erros - Node Guarding

Este serviço é utilizado para possibilitar a monitoração da comunicação com a rede CANopen, tanto pelo mestre quanto pelo escravo. Neste tipo de serviço, o mestre envia telegramas periódicos para o escravo, que responde o telegrama recebido. Caso ocorra algum erro que interrompa a comunicação, será possível identificar este erro, pois tanto o mestre quanto o escravo serão notificados pelo timeout na execução deste serviço. Os eventos de erro são chamados de Node Guarding para o mestre, e de Life Guarding para o escravo.


**Figura 8.5:** Serviço de controle de erros – Node Guarding

Para o serviço de Node Guarding, existem dois objetos do dicionário para configuração dos tempos para detecção de erros de comunicação:

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
100Ch	0	Guard Time	UNSIGNED16	RW	No	0

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
100Dh	0	Life Time Factor	UNSIGNED8	RW	No	0

O objeto 100Ch permite programar o tempo necessário (em milissegundos) para que uma ocorrência de falha seja detectada, caso o escravo não receba nenhum telegrama do mestre. O objeto 100Dh indica quantas falhas em sequência são necessárias até que se considere que houve realmente perda da comunicação. Portanto, a

multiplicação destes dois valores fornecerá o tempo total necessário para detecção de erros de comunicação utilizando este objeto. O valor 0 (zero) desabilita esta função.

Uma vez configurado, o escravo começa a contar estes tempos a partir do primeiro telegrama Node Guarding recebido do mestre da rede. O telegrama do mestre é do tipo remoto, não possuindo bytes de dados. O identificador é igual a 700h + Node-ID do escravo destino. Já o telegrama de resposta do escravo possui 1 byte de dados com a seguinte estrutura:

Identificador	byte 1	
	bit 7	bit 6 ... 0
700h + Node ID	Toogle	Estado do Escravo

Este telegrama possui um único byte dados. Este byte contém, nos sete bits menos significativos, um valor para indicar o estado do escravo (4 = Parado, 5 = Operacional e 127 = Pré-operacional), e no oitavo bit, um valor que deve ser alterado a cada telegrama enviado pelo escravo (toggle bit).

Caso o gerenciador de partidas SCW100 detecte um erro utilizando este mecanismo, irá automaticamente para o estado pré-operacional e indicará com o LED de erro.

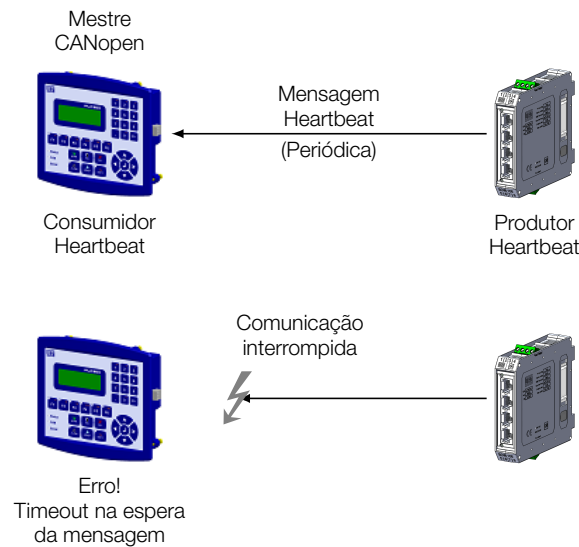

**NOTA!**

- Este objeto está ativo mesmo no estado parado (consulte a tabela 8.11).
- O valor 0 (zero) em um dos dois objetos desabilita esta função.
- Depois de detectado o erro, caso o serviço seja habilitado mais uma vez, a indicação do erro é retirada.
- O valor mínimo aceito para o gerenciador de partidas SCW100 é de 2 ms. Mas levando-se em conta a taxa de transmissão e o número de pontos na rede, os tempos programados para essa função devem ser coerentes, de maneira que haja tempo suficiente para transmissão dos telegramas e também para que o resto da comunicação possa ser processada.
- Para cada escravo, somente um dos serviços – Heartbeat ou Node Guarding – pode ser habilitado.

### 8.5.3 Controle de Erros - Heartbeat

A detecção de erros através do mecanismo de heartbeat é feita utilizando dois tipos de objetos: o produtor heartbeat e o consumidor heartbeat. O produtor é responsável por enviar telegramas periódicos para a rede, simulando uma batida do coração, indicando que a comunicação está ativa e sem erros. Um ou mais consumidores podem monitorar estes telegramas periódicos e, caso estes telegramas deixem de ocorrer, significa que algum problema de comunicação ocorreu.





**Figura 8.6:** Serviço de controle de erros – Heartbeat

Um mesmo dispositivo da rede pode ser produtor e consumidor de mensagens heartbeat. Por exemplo, o mestre da rede pode consumir mensagens enviadas por um escravo, permitindo detectar problemas de comunicação com o escravo, e ao mesmo tempo o escravo pode consumir mensagens heartbeat enviadas pelo mestre, também possibilitando ao escravo detectar falhas na comunicação com o mestre.

O gerenciador de partidas SCW100 possui os serviços de produtor e consumidor heartbeat. Como consumidor, é possível programar diferentes produtores para serem monitorados pelo equipamento:

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1016h	0	Número do último sub-índice	UNSIGNED8	RO	No	4
	1 até 4	Consumer Heartbeat Time 1 até 4	UNSIGNED32	RW	No	0

Nos sub-índices de 1 até 4, é possível programar o consumidor escrevendo um valor no seguinte formato:

UNSIGNED32		
Reservado (8 bits)	Node-ID (8 bits)	HeartBeat time (16 bits)

- Node-ID: permite programar o Node-ID do produtor heartbeat o qual se deseja monitorar.
- Heartbeat time: permite programar o tempo, em múltiplos de 1 milissegundo, até a detecção de erro, caso nenhuma mensagem do produtor seja recebida. O valor 0 (zero) neste campo desabilita o consumidor.

Depois de configurado, o consumidor heartbeat inicia a monitoração após o primeiro telegrama enviado pelo produtor. Caso seja detectado erro pelo fato do consumidor deixar de receber mensagens do produtor heartbeat, este irá automaticamente para o estado pré-operacional e indicará com o LED de erro.

Como produtor, o gerenciador de partidas SCW100 possui um objeto para configuração deste serviço:

Índice	Sub-índice	Nome	Tipo	Acesso	PDO Mapping	Valor
1017h	0	Producer Heartbeat Time	UNSIGNED16	RW	No	0

O objeto 1017h permite programar o tempo em milissegundos no qual o produtor envie um telegrama heartbeat para a rede. Uma vez programado, o dispositivo inicia a transmissão de mensagens com o seguinte formato:

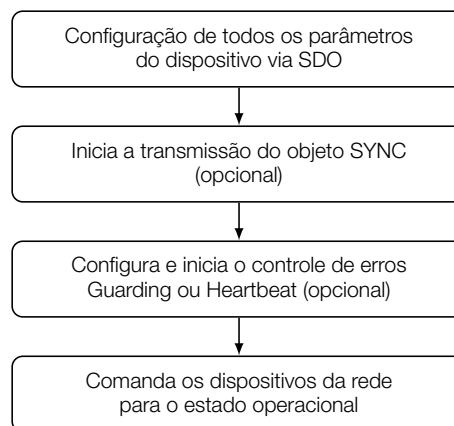
Identificador	byte 1	
	bit 7	bit 6 ... 0
700h + Node ID	Sempre 0	Estado do Escravo


**NOTA!**

- Este objeto está ativo mesmo no estado parado (consulte a tabela 8.11).
- O valor 0 (zero) em um dos dois objetos desabilita esta função.
- Depois de detectado o erro, caso o serviço seja habilitado mais uma vez, a indicação do erro é retirado.
- O valor mínimo aceito para o gerenciador de partidas SCW100 é de 2 ms. Mas levando-se em conta a taxa de transmissão e o número de pontos na rede, os tempos programados para essa função devem ser coerentes, de maneira que haja tempo suficiente para transmissão dos telegramas e também para que o resto da comunicação possa ser processada.
- Para cada escravo, somente um dos serviços – Heartbeat ou Node Guarding – pode ser habilitado.

## 8.6 PROCEDIMENTO DE INICIALIZAÇÃO

Uma vez conhecido o funcionamento dos objetos disponíveis para o gerenciador de partidas SCW100 operando no modo escravo, é necessário agora programar os diferentes objetos para operarem em conjunto na rede. De forma geral, o procedimento para inicialização dos objetos em uma rede CANopen segue o descrito pelo fluxograma a seguir:



**Figura 8.7:** Fluxograma do processo de inicialização

É necessário observar que os objetos de comunicação do gerenciador de partidas SCW100 (1000h até 1FFFh) não são armazenados na memória não volátil. Desta forma, sempre que for feito o reset ou desligado o equipamento, é necessário refazer a parametrização dos objetos de comunicação. Para os objetos específicos do fabricante (a partir de 2000h, que representam os parâmetros), estes são armazenados na memória não volátil e, portanto, podem ser parametrizados uma única vez.

## 9 COLOCAÇÃO EM OPERAÇÃO

A seguir são descritos os principais passos para colocação em funcionamento do gerenciador de partidas SCW100 em rede CANopen. Os passos descritos representam um exemplo de uso. Consulte os capítulos específicos para detalhes sobre os passos indicados.

### 9.1 INSTALAÇÃO DO PRODUTO NA REDE

1. Instale o gerenciador de partidas SCW100 na rede CANopen e faça as configurações necessárias para a operação conforme descrito no item 2.
2. Conecte os cabos, considerando os cuidados necessários na instalação da rede, conforme descrito no item 3.5:
  - Utilize cabo blindado.
  - Aterre adequadamente os equipamentos da rede.
  - Evite a passagem dos cabos de comunicação próximos aos cabos de potência.

### 9.2 CONFIGURAÇÃO DO EQUIPAMENTO

1. Seguir as recomendações descritas no manual do usuário para programar parâmetros de ajuste do equipamento.
2. Programar ajustes de comunicação, como endereço e taxa de comunicação nas chave DIP switch S2.
3. Programar a ação desejada para o equipamento em caso de falha na comunicação, através do parâmetro P0101: Ação para Falha na Comunicação.

### 9.3 CONFIGURAÇÃO DO MESTRE

A forma como é feita a configuração da rede depende muito do mestre utilizado e da ferramenta de configuração. É fundamental conhecer as ferramentas utilizadas para realizar esta atividade. De uma maneira geral, os seguintes passos são necessários para realizar a configuração da rede.

1. Carregue o arquivo de configuração EDS<sup>3</sup> para a lista de equipamentos na ferramenta de configuração da rede.
2. Selecione o gerenciador de partidas SCW100 na lista de equipamentos disponíveis no configurador da rede. Isto pode ser feito manualmente ou de forma automática, se a ferramenta permitir.
3. Durante a configuração da rede, é necessário definir quais dados serão lidos e escritos no gerenciador de partidas SCW100 configurando os PDOs de transmissão e recepção conforme descrito no item 8.3.
4. Configurar o controle de erros utilizando os serviços Node Guarding ou Heartbeat descritos no item 8.5.

Uma vez configurado, o estado da rede em P0038: CAN - Estado da Comunicação CANopen indicará 2 ou 3 e o estado do nó em P0039: CAN - Estado do Escravo CANopen indicará 3. É nesta condição que ocorre efetivamente a transmissão e recepção dos PDOs.

<sup>3</sup>O arquivo de configuração EDS está disponível na página de internet da WEG (<http://www.weg.net>). É importante observar se o arquivo de configuração EDS é compatível com a versão de firmware do gerenciador de partidas SCW100.

## 9.4 ESTADO DA COMUNICAÇÃO

Uma vez que a rede esteja montada e o mestre programado, é possível utilizar o LED Net e parâmetros do equipamento para identificar alguns estados relacionados com a comunicação.

- O LED Net fornecem informações sobre o estado da interface.
- Os parâmetros P0038: CAN - Estado da Comunicação CANopen e P0039: CAN - Estado do Escravo CANopen indicam o estado da comunicação CANopen.

O mestre da rede também deve fornecer informações sobre a comunicação com o escravo.

## 9.5 OPERAÇÃO UTILIZANDO DADOS DE PROCESSO

Uma vez que a comunicação esteja estabelecida, os dados mapeados nos PDOs são automaticamente atualizados.

É importante conhecer estes parâmetros para programar o mestre conforme desejado para a aplicação.

## 9.6 ACESSO AOS PARÂMETROS – MENSAGENS ACÍCLICAS

Além da comunicação cíclica utilizando os PDOs, o protocolo CANopen também define um tipo de mensagem acíclica via SDO, utilizado principalmente em tarefas assíncronas tais como parametrização e configuração do equipamento.

O arquivo EDS possui a listagem completa dos parâmetros do equipamento os quais podem ser acessados via SDO. O item 7.4 descreve como endereçar os parâmetros do gerenciador de partidas SCW100 para acesso via SDO.

## 10 FALHAS E ALARMES

Falha/Alarme	Descrição	Causas Mais Prováveis
Bus Off	<p>Detectado erro de bus off na interface CAN. Caso o número de erros de recepção ou transmissão detectados pela interface CAN seja muito elevado, o controlador CAN pode ser levado ao estado de bus off, onde ele interrompe a comunicação e desabilita a interface CAN.</p> <p>Para que a comunicação seja restabelecida, é necessário desligar e ligar novamente o produto para que a comunicação seja reiniciada.</p> <p>Neste caso será sinalizado através do LED Net no gerenciador de partidas. A comunicação é restabelecida automaticamente se o parâmetro P0032 - CAN - Reset de Bus Off estiver com valor 1. Se o valor do parâmetro P0032 - CAN - Reset de Bus Off for 0, será necessário desligar e ligar a unidade remota para restabelecer a comunicação.</p>	<ul style="list-style-type: none"> <li>- Verificar curto-circuito nos cabos de transmissão do circuito CAN.</li> <li>- Verificar se os cabos não estão trocados ou invertidos.</li> <li>- Verificar se todos os dispositivos da rede utilizam a mesma taxa de comunicação.</li> <li>- Verificar se resistores de terminação com valores corretos foram colocados somente nos extremos do barramento principal.</li> <li>- Verificar se a instalação da rede CAN foi feita de maneira adequada.</li> </ul>
CANopen Offline	<p>Ocorre caso o estado do nó CANopen passe de operacional para pré-operacional.</p>	<ul style="list-style-type: none"> <li>- Verificar o funcionamento dos mecanismos de controle de erros (Heartbeat/Node Guarding).</li> <li>- Verificar se o mestre está enviando os telegramas de guarding/heartbeat no tempo programado.</li> <li>- Verificar problemas na comunicação que possam ocasionar perda de telegramas ou atrasos na transmissão.</li> </ul>

## APÊNDICE A REFERÊNCIA RÁPIDA DOS PARÂMETROS

Nível 1	Pág.
S	Status
S1	Informações do Produto 39
S2	Partidas 39
S3	Erros e Alarmes 40
S4	Comunicação 41
C	Configurações
C1	Partidas 42
C2	Comunicação 42
C3	Contadores 42
C4	Comandos 42

Tabela A.2: Referência rápida dos parâmetros

Parâmetro	Descrição	Faixa de Valores	Ajuste de fábrica	Propriedades	Endereço Comunicação	CANopen Index	Sub-Index
<b>S1 Status\Informações do Produto</b>							
P0001	Versão de Firmware	0,0 a 655,35	-	ro, 16bit	1	2001h	0
P0010	Chave Rotativa S1	0 a 15	-	ro, 8bit	10	200Ah	0
P0011	Chave Rotativa S2	0 a 15	-	ro, 8bit	11	200Bh	0
P0012	Chave Rotativa S3	0 a 15	-	ro, 8bit	12	200Ch	0
P0015	Estados das Entradas Digitais	Bit 0 = DI1 Bit 1 = DI2 Bit 2 = DI3 Bit 3 = DI4 Bit 4 = DI5 Bit 5 = DI6 Bit 6 = DI7 Bit 7 = DI8 Bit 8 = DI9 Bit 9 = DI10 Bit 10 = DI11 Bit 11 = DI12	-	ro, 16bit	15	200Fh	0
P0016	Estados das Saídas Digitais	Bit 0 = DO1 Bit 1 = DO2 Bit 2 = DO3 Bit 3 = DO4 Bit 4 = DO5 Bit 5 = DO6 Bit 6 = DO7 Bit 7 = DO8	-	ro, 16bit	16	2010h	0
P0500	Temperatura da CPU	-100 a 100 °C	-	ro, s16bit	500	21F4h	0
<b>S2 Status\Partidas</b>							
P0120	P1 Tempo de fechamento Contator 1	0 a 65535 ms	-	ro, 16bit	120	2078h	0
P0121	P1 Tempo de abertura Contator 1	0 a 65535 ms	-	ro, 16bit	121	2079h	0
P0122	P1 Tempo de fechamento Contator 2	0 a 65535 ms	-	ro, 16bit	122	207Ah	0
P0123	P1 Tempo de abertura Contator 2	0 a 65535 ms	-	ro, 16bit	123	207Bh	0
P0124	P2 Tempo de fechamento Contator 1	0 a 65535 ms	-	ro, 16bit	124	207Ch	0
P0125	P2 Tempo de abertura Contator 1	0 a 65535 ms	-	ro, 16bit	125	207Dh	0
P0126	P2 Tempo de fechamento Contator 2	0 a 65535 ms	-	ro, 16bit	126	207Eh	0
P0127	P2 Tempo de abertura Contator 2	0 a 65535 ms	-	ro, 16bit	127	207Fh	0
P0128	P3 Tempo de fechamento Contator 1	0 a 65535 ms	-	ro, 16bit	128	2080h	0
P0129	P3 Tempo de abertura Contator 1	0 a 65535 ms	-	ro, 16bit	129	2081h	0
P0130	P3 Tempo de fechamento Contator 2	0 a 65535 ms	-	ro, 16bit	130	2082h	0
P0131	P3 Tempo de abertura Contator 2	0 a 65535 ms	-	ro, 16bit	131	2083h	0
P0132	P4 Tempo de fechamento Contator 1	0 a 65535 ms	-	ro, 16bit	132	2084h	0
P0133	P4 Tempo de abertura Contator 1	0 a 65535 ms	-	ro, 16bit	133	2085h	0
P0134	P4 Tempo de fechamento Contator 2	0 a 65535 ms	-	ro, 16bit	134	2086h	0
P0135	P4 Tempo de abertura Contator 2	0 a 65535 ms	-	ro, 16bit	135	2087h	0



Parâmetro	Descrição	Faixa de Valores	Ajuste de fábrica	Propriedades	Endereço Comunicação	CANopen Index	Sub-Index
P0150	Contador de manobras P1 C1	0 a 65535	-	ro, 16bit	150	2096h	0
P0151	Contador de manobras P1 C2	0 a 65535	-	ro, 16bit	151	2097h	0
P0152	Contador de manobras P2 C1	0 a 65535	-	ro, 16bit	152	2098h	0
P0153	Contador de manobras P2 C2	0 a 65535	-	ro, 16bit	153	2099h	0
P0154	Contador de manobras P3 C1	0 a 65535	-	ro, 16bit	154	209Ah	0
P0155	Contador de manobras P3 C2	0 a 65535	-	ro, 16bit	155	209Bh	0
P0156	Contador de manobras P4 C1	0 a 65535	-	ro, 16bit	156	209Ch	0
P0157	Contador de manobras P4 C2	0 a 65535	-	ro, 16bit	157	209Dh	0
P0200	Status P1	0 a 65535	-	ro, 16bit	200	20C8h	0
P0201	Status P1 - Contator	0 a 1	-	ro, 16bit	201	20C9h	0
P0202	Status P1 - Erro	0 a 65535	-	ro, 16bit	202	20CAh	0
P0203	Status P1 - Alarme	0 a 65535	-	ro, 16bit	203	20CBh	0
P0204	Status P2	0 a 65535	-	ro, 16bit	204	20CCh	0
P0205	Status P2 - Contator	0 a 1	-	ro, 16bit	205	20CDh	0
P0206	Status P2 - Erro	0 a 65535	-	ro, 16bit	206	20CEh	0
P0207	Status P2 - Alarme	0 a 65535	-	ro, 16bit	207	20CFh	0
P0208	Status P3	0 a 65535	-	ro, 16bit	208	20D0h	0
P0209	Status P3 - Contator	0 a 1	-	ro, 16bit	209	20D1h	0
P0210	Status P3 - Erro	0 a 65535	-	ro, 16bit	210	20D2h	0
P0211	Status P3 - Alarme	0 a 65535	-	ro, 16bit	211	20D3h	0
P0212	Status P4	0 a 65535	-	ro, 16bit	212	20D4h	0
P0213	Status P4 - Contator	0 a 1	-	ro, 16bit	213	20D5h	0
P0214	Status P4 - Erro	0 a 65535	-	ro, 16bit	214	20D6h	0
P0215	Status P4 - Alarme	0 a 65535	-	ro, 16bit	215	20D7h	0
<b>S3 Status\Erros e Alarmes</b>							
P0300	P1 - Último Erro 1	0 a 65535	-	ro, 16bit	300	212Ch	0
P0301	P1 - Último Erro 2	0 a 65535	-	ro, 16bit	301	212Dh	0
P0302	P1 - Último Erro 3	0 a 65535	-	ro, 16bit	302	212Eh	0
P0303	P1 - Último Erro 4	0 a 65535	-	ro, 16bit	303	212Fh	0
P0304	P1 - Último Erro 5	0 a 65535	-	ro, 16bit	304	2130h	0
P0305	P2 - Último Erro 1	0 a 65535	-	ro, 16bit	305	2131h	0
P0306	P2 - Último Erro 2	0 a 65535	-	ro, 16bit	306	2132h	0
P0307	P2 - Último Erro 3	0 a 65535	-	ro, 16bit	307	2133h	0
P0308	P2 - Último Erro 4	0 a 65535	-	ro, 16bit	308	2134h	0
P0309	P2 - Último Erro 5	0 a 65535	-	ro, 16bit	309	2135h	0
P0310	P3 - Último Erro 1	0 a 65535	-	ro, 16bit	310	2136h	0
P0311	P3 - Último Erro 2	0 a 65535	-	ro, 16bit	311	2137h	0
P0312	P3 - Último Erro 3	0 a 65535	-	ro, 16bit	312	2138h	0
P0313	P3 - Último Erro 4	0 a 65535	-	ro, 16bit	313	2139h	0
P0314	P3 - Último Erro 5	0 a 65535	-	ro, 16bit	314	213Ah	0
P0315	P4 - Último Erro 1	0 a 65535	-	ro, 16bit	315	213Bh	0
P0316	P4 - Último Erro 2	0 a 65535	-	ro, 16bit	316	213Ch	0
P0317	P4 - Último Erro 3	0 a 65535	-	ro, 16bit	317	213Dh	0
P0318	P4 - Último Erro 4	0 a 65535	-	ro, 16bit	318	213Eh	0



Parâmetro	Descrição	Faixa de Valores	Ajuste de fábrica	Propriedades	Endereço Comunicação	CANopen Index	Sub-Index
P0319	P4 - Último Erro 5	0 a 65535	-	ro, 16bit	319	213Fh	0
P0320	P1 - Último Alarme 1	0 a 65535	-	ro, 16bit	320	2140h	0
P0321	P1 - Último Alarme 2	0 a 65535	-	ro, 16bit	321	2141h	0
P0322	P1 - Último Alarme 3	0 a 65535	-	ro, 16bit	322	2142h	0
P0323	P1 - Último Alarme 4	0 a 65535	-	ro, 16bit	323	2143h	0
P0324	P1 - Último Alarme 5	0 a 65535	-	ro, 16bit	324	2144h	0
P0325	P2 - Último Alarme 1	0 a 65535	-	ro, 16bit	325	2145h	0
P0326	P2 - Último Alarme 2	0 a 65535	-	ro, 16bit	326	2146h	0
P0327	P2 - Último Alarme 3	0 a 65535	-	ro, 16bit	327	2147h	0
P0328	P2 - Último Alarme 4	0 a 65535	-	ro, 16bit	328	2148h	0
P0329	P2 - Último Alarme 5	0 a 65535	-	ro, 16bit	329	2149h	0
P0330	P3 - Último Alarme 1	0 a 65535	-	ro, 16bit	330	214Ah	0
P0331	P3 - Último Alarme 2	0 a 65535	-	ro, 16bit	331	214Bh	0
P0332	P3 - Último Alarme 3	0 a 65535	-	ro, 16bit	332	214Ch	0
P0333	P3 - Último Alarme 4	0 a 65535	-	ro, 16bit	333	214Dh	0
P0334	P3 - Último Alarme 5	0 a 65535	-	ro, 16bit	334	214Eh	0
P0335	P4 - Último Alarme 1	0 a 65535	-	ro, 16bit	335	214Fh	0
P0336	P4 - Último Alarme 2	0 a 65535	-	ro, 16bit	336	2150h	0
P0337	P4 - Último Alarme 3	0 a 65535	-	ro, 16bit	337	2151h	0
P0338	P4 - Último Alarme 4	0 a 65535	-	ro, 16bit	338	2152h	0
P0339	P4 - Último Alarme 5	0 a 65535	-	ro, 16bit	339	2153h	0
<b>S4 Status\Comunicação</b>							
P0030	CAN - Endereço	1 a 127	-	ro, 16bit	30	201Eh	0
P0031	CAN - Taxa de Comunicação	0 = 1 Mbit/s 1 = 800 Kbit/s 2 = 500 Kbit/s 3 = 250 Kbit/s 4 = 125 Kbit/s 5 = 100 Kbit/s 6 = 50 Kbit/s 7 = 20 Kbit/s	-	ro, enum	31	201Fh	0
P0033	CAN - Estado do Controlador	0 = Inativo 1 = Reservado 2 = CAN Ativo 3 = Warning 4 = Erro Passivo 5 = Bus Off	-	ro, enum	33	2021h	0
P0034	CAN - Telegramas CAN RX	0 a 65535	-	ro, 16bit	34	2022h	0
P0035	CAN - Telegramas CAN TX	0 a 65535	-	ro, 16bit	35	2023h	0
P0036	CAN - Contador Bus Off	0 a 65535	-	ro, 16bit	36	2024h	0
P0037	CAN - Telegramas Perdidos	0 a 65535	-	ro, 16bit	37	2025h	0
P0038	CAN - Estado da Comunicação CANopen	0 = Inativo 1 = Reservado	-	ro, enum	38	2026h	0

Parâmetro	Descrição	Faixa de Valores	Ajuste de fábrica	Propriedades	Endereço Comunicação	CANopen Index	Sub-Index
P0039	CAN - Estado do Escravo CANopen	2 = Comunic. Hab. 3 = Ctrl. Erros Hab. 4 = Erro Guarding 5 = Erro Heartbeat  0 = Inativo 1 = Inicialização 2 = Parado 3 = Operacional 4 = PreOperacional	-	ro, enum	39	2027h	0
<b>C1 Configurações\Partidas</b>							
P0400	P1 - Modo de Operação	0 = Partida 1 = Transparente	0	rw, 16bit	400	2190h	0
P0401	P2 - Modo de Operação	0 = Partida 1 = Transparente	0	rw, 16bit	401	2191h	0
P0402	P3 - Modo de Operação	0 = Partida 1 = Transparente	0	rw, 16bit	402	2192h	0
P0403	P4 - Modo de Operação	0 = Partida 1 = Transparente	0	rw, 16bit	403	2193h	0
P0404	P1 - Timeout Contator	20 a 5000 ms	500 ms	rw, 16bit	404	2194h	0
P0405	P2 - Timeout Contator	20 a 5000 ms	500 ms	rw, 16bit	405	2195h	0
P0406	P3 - Timeout Contator	20 a 5000 ms	500 ms	rw, 16bit	406	2196h	0
P0407	P4 - Timeout Contator	20 a 5000 ms	500 ms	rw, 16bit	407	2197h	0
P1000	Reseta Padrão de Fábrica	0 a 65535	0	rw, 16bit	1000	23E8h	0
<b>C2 Configurações\Comunicação</b>							
P0101	Ação para Falha na Comunicação	0 = Sem Ação 1 = Desliga Saídas	1	rw, enum	101	2065h	0
P0032	CAN - Reset de Bus Off	0 = Manual 1 = Automático	0	rw, enum	32	2020h	0
<b>C3 Configurações\Contadores</b>							
P0158	Salva Contadores de manobras na mem NV	FALSE a TRUE	FALSE	rw, bool	158	209Eh	0
P0160	Reseta Contador de manobras P1 C1	0 a 65535	0	rw, 16bit	160	20A0h	0
P0161	Reseta Contador de manobras P1 C2	0 a 65535	0	rw, 16bit	161	20A1h	0
P0162	Reseta Contador de manobras P2 C1	0 a 65535	0	rw, 16bit	162	20A2h	0
P0163	Reseta Contador de manobras P2 C2	0 a 65535	0	rw, 16bit	163	20A3h	0
P0164	Reseta Contador de manobras P3 C1	0 a 65535	0	rw, 16bit	164	20A4h	0
P0165	Reseta Contador de manobras P3 C2	0 a 65535	0	rw, 16bit	165	20A5h	0
P0166	Reseta Contador de manobras P4 C1	0 a 65535	0	rw, 16bit	166	20A6h	0
P0167	Reseta Contador de manobras P4 C2	0 a 65535	0	rw, 16bit	167	20A7h	0
<b>C4 Configurações\Comandos</b>							

Parâmetro	Descrição	Faixa de Valores	Ajuste de fábrica	Propriedades	Endereço Comunicação	CANopen Index	Sub-Index
P1100	Comando de Partida Direta	Bit 0 = Partida 1 - direto Bit 1 = Partida 2 - direto Bit 2 = Partida 3 - direto Bit 3 = Partida 4 - direto	0	rw, 16bit	1100	244Ch	0
P1105	Comando de Partida Reversa	Bit 0 = Partida 1 - reverso Bit 1 = Partida 2 - reverso Bit 2 = Partida 3 - reverso Bit 3 = Partida 4 - reverso	0	rw, 16bit	1105	2451h	0
P1110	Comando de Parada	Bit 0 = Partida 1 - desliga Bit 1 = Partida 2 - desliga Bit 2 = Partida 3 - desliga Bit 3 = Partida 4 - desliga	0	rw, 16bit	1110	2456h	0
P1115	Comando das Saídas Digitais	Bit 0 = DO1 Bit 1 = DO2 Bit 2 = DO3 Bit 3 = DO4 Bit 4 = DO5 Bit 5 = DO6 Bit 6 = DO7 Bit 7 = DO8	0	rw, 16bit	1115	245Bh	0

**Tabela A.3:** Descrição dos tipos de dados dos parâmetros

Tipo de Dado	Descrição
bool	Representa um bit onde o valor 0 (zero) representa falso, o valor 1 (um) representa verdadeiro.
enum	Tipo enumerado (8 bits sem sinal), contém uma lista de valores com descrição da função para cada item.
8bit	Inteiro de 8 bits sem sinal, varia de 0 a 255.
16bit	Inteiro de 16 bits sem sinal, varia de 0 a 65.535.
s16bit	Inteiro de 16 bits com sinal, varia de -32.768 a 32.767.
32bit	Inteiro de 32 bits sem sinal, varia de 0 a 4.294.967.295.



WEG Drives & Controls - Automação LTDA.  
Jaraguá do Sul – SC – Brasil  
Fone 55 (47) 3276-4000 – Fax 55 (47) 3276-4020  
São Paulo – SP – Brasil  
Fone 55 (11) 5053-2300 – Fax 55 (11) 5052-4212  
[automacao@weg.net](mailto:automacao@weg.net)  
[www.weg.net](http://www.weg.net)