

Driver Elipse SuperDriver

Nome do Arquivo	SuperDriver.dll
Fabricante	Elipse Software
Equipamentos	Qualquer equipamento que comunique com Drivers da Elipse Software desenvolvidos com a biblioteca IOKit e que possam comunicar em Ethernet TCP/IP
Protocolo	Qualquer protocolo sobre TCP/IP implementado em Drivers da Elipse Software desenvolvidos com a biblioteca IOKit versão 2.0 e que possam comunicar em Ethernet TCP/IP
Versão	2.1.10
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	N/A
Leitura com Superblocos	Não
Nível	31302

Introdução

O objetivo primário do Driver Elipse SuperDriver é comunicar através de outros Drivers da **Elipse Software**, podendo cada um destes Drivers ser especializado em algum protocolo. Múltiplos Drivers são associados ao Driver Elipse SuperDriver, gerando itens chamados **Escravos**, que têm definições de configurações isoladas por **Perfis de Drivers**. Assim, os Tags de Comunicação existentes em uma aplicação, referenciados a estes **Escravos** por algum campo de configuração, têm os parâmetros repassados à respectiva instância de Driver apontada.

Os Drivers da **Elipse Software** que podem ser subordinados ao Driver Elipse SuperDriver devem ser aqueles desenvolvidos para a biblioteca IOKit, e a topologia deve proporcionar obrigatoriamente como camada física de comunicação imediata à aplicação a interface **Ethernet**, e como protocolo de transporte e rede o **TCP/IP**.

O Driver Elipse SuperDriver é incompatível com o **Elipse SCADA**.

Funcionamento do Driver

Para que o Driver Elipse SuperDriver seja utilizado adequadamente em uma aplicação, é realmente importante entender como este Driver funciona.

Definições

É necessário familiarizar-se antes com alguns conceitos que são utilizados mais adiante neste documento, para que o funcionamento do Driver Elipse SuperDriver seja bem entendido.

Perfil de Driver

Perfil de Driver é um objeto de uma estrutura de dados que contém um nome e um caminho para um arquivo da biblioteca de ligação dinâmica (DLL) de um Driver da **Elipse Software**, além de uma declaração de configurações para este Driver. Vários Perfis de Drivers podem compartilhar o mesmo arquivo de Driver, cada um com uma configuração e um nome diferente.

Na aba de propriedades SuperDriver há uma lista de declarações de Perfis, que é explicada em capítulo posterior. Para que o Driver Elipse SuperDriver funcione, uma lista de Perfis de Drivers deve ser configurada com pelo menos 1 (um) item.

Status de um Tag em Advise

O status **Em Advise** de um Tag é definido para que este Tag se mantenha em atualização constante, em intervalos de tempo definidos pela propriedade **Scan**, ou períodos de varredura.

A forma de mudar o status de um Tag para **Em Advise** é configurando a propriedade **AllowRead** para Verdadeiro e por uma das seguintes condições:

- Um Tag está com o valor da propriedade **AdviseType** igual a 0 (zero), ou seja, igual a **AlwaysInAdvise**
- Um Tag está com o valor da propriedade **AdviseType** igual a 1 (um), ou seja, igual a **AdviseWhenLinked**, e está associado a algum objeto ativo em uma Tela aberta

Caso alguma destas condições não esteja presente, um Tag de Comunicação não tem o status **Em Advise**, ou seja, não é lido ou escrito.

No caso de um serviço de escrita através do Driver Elipse SuperDriver, não basta apenas habilitar a propriedade **AllowWrite** de um Tag, é preciso também que haja pelo menos um Tag de Comunicação com o status **Em Advise**, conforme a definição anterior, ativo no respectivo Perfil de Driver.

É somente nesta condição que um Tag de escrita pode ser adicionado na lista de *cache* de um **Escravo** para posterior processamento dentro dos períodos de varredura.

Portanto, não é possível realizar a escrita de um Tag de forma direta ou individualizada. Uma escrita sempre é processada via lista de *cache* de um **Escravo**, em conjunto com os demais Tags **Em Advise**.

Escravo

Entende-se por **Escravo** um objeto criado dinamicamente pelo Driver Elipse SuperDriver, e que é portador de um *cache* (repositório) de Tags e de recursos de memória independentes, e relacionado a um Perfil de Driver de Comunicação. Os **Escravos** são criados pelo Driver Elipse SuperDriver a partir do fornecimento de pares de Perfis e Portas pelos Tags declarados.

O par **Perfil** e **Porta** é fornecido pelo conteúdo do campo **Dispositivo**. O campo **Dispositivo** de cada Tag deve conter o nome de um dos Perfis declarados e um número de porta TCP/IP, separados pelo caractere de dois pontos. Esta sintaxe é apresentada mais adiante.

Existe um **Escravo** para cada par destes diferente que tenha sido declarado no conjunto de Tags subordinados ao objeto do Driver Elipse SuperDriver em uma aplicação. Dez mil **Escravos** é o valor máximo suportado pelo Driver Elipse SuperDriver.

Modos de Operação

O Driver Elipse SuperDriver pode operar nos modos Varredura Geral (*General Scan*) ou Varredura Individual (*Individual Scan*). O modo **Varredura Geral** é executado sempre, para todos Escravos que possuem Tags **Em Advise**, e o modo **Varredura Individual** de um **Escravo** somente é executado se está habilitado.

Varredura Geral

O Driver Elipse SuperDriver usa o modo **Varredura Geral**, o tempo de varredura geral existente na aba de configurações, para realizar a leitura e escrita de todos os Tags **Em Advise** de todos os **Escravos**. Cada vez que o modo **Varredura Geral** de cada **Escravo** expira, este inicia a leitura e escrita de todos os Tags, obedecendo à seguinte estratégia de conexão:

1. O **Escravo** inicia o respectivo objeto de Driver, sem habilitar a conexão automática a um endereço IP de *backup*.
2. O **Escravo** tenta processar todos os Tags. À medida que os Tags vão tendo sucesso, estes são retirados da lista de *cache*.
3. Ao final de uma iteração de serviços, o objeto de Driver é parado e deixa-se o recurso disponível, ou seja, a porta TCP/IP, caso haja um outro **Escravo** querendo este recurso.
4. Existindo Tags na lista de *cache* ao final de uma primeira iteração de serviços, a leitura ou escrita é retentada até *n* vezes, conforme o número de retentativas configurado na aba de propriedades do Driver Elipse SuperDriver.
5. Se ainda há Tags na lista de *cache* ao final de todas as retentativas, e o número de Tags de Comunicação permanece igual ao número inicialmente requisitado, o Driver Elipse SuperDriver toma o endereço IP de *backup* como endereço IP principal e retenta novamente o ciclo de serviços com retentativas por este endereço IP. Isto evidentemente está condicionado ao que foi configurado na aba de propriedades. Se o usuário não deseja um endereço IP de *backup*, não há esta retentativa.
6. Se o número de Tags de Comunicação é inferior ao número inicialmente requisitado, isto é, se obteve sucesso em pelo menos um Tag, o processamento termina sem tentar o endereço IP de *backup*.

NOTA

Se há uma recusa de conexão por parte do endereço IP de destino, tanto para o endereço IP principal quanto para o endereço IP de *backup*, considera-se que um equipamento está em estado desligado, e portanto não são realizadas as retentativas programadas.

Varredura Individual

Assim como há o modo **Varredura Geral**, todos os Tags dos **Escravos** têm os *scans* individuais respeitados se existe **Em Advise** de um Tag **Varredura Individual** para o respectivo Perfil. As leituras e escritas realizadas por estes tempos não influenciam nos tempos de leitura e escrita do modo **Varredura Geral**, isto é, o serviço por **Varredura Individual** não reinicia o tempo que falta para o serviço por **Varredura Geral**. Este *scan* individual sem influência sobre o modo **Varredura Geral** é desejado para leituras e escritas provocadas ao longo do intervalo entre *scans* gerais, que dessincronizam a expiração de todos os Tags ao mesmo tempo. O modo **Varredura Geral** cuida, portanto, da atualização de todos os Tags **Em Advise** de todos os Perfis em ocorrências entre intervalos iguais para todos. Se há um ou mais Tags com *scan* individual expirado, estes são agrupados e colocados em uma lista de *cache*, obedecendo à seguinte estratégia de conexão:

1. O **Escravo** inicia o respectivo objeto de Driver, habilitando a conexão automática a um endereço IP de *backup*.
2. O módulo **IOKit** do respectivo objeto de Driver do **Escravo** se encarrega de automaticamente tentar um endereço IP principal ou um endereço IP de *backup*, de acordo com a configuração do gerenciamento de conexão automática.

3. O **Escravo** realiza o processamento de todos os Tags da lista de *cache*. À medida que os Tags vão tendo sucesso, estes são retirados desta lista.
4. Esta lista de Tags em **Varredura Individual** também retenta n vezes, conforme configurado na aba de propriedades, enquanto há Tags na lista de *cache*. A cada retentativa, o Perfil libera o recurso caso exista outro Perfil concorrente a este.
5. O processamento dos Tags individualizados termina com o sucesso de todo o grupo ou ao atingir o final das retentativas.

NOTA

Se há uma recusa de conexão por parte do endereço IP de destino, tanto para o endereço IP principal quanto para o endereço IP de *backup*, considera-se que um equipamento está em estado desligado, e portanto não são realizadas as retentativas programadas.

Execução

Quando colocado em execução, o Driver Elipse SuperDriver imediatamente cria os objetos **Escravos**. No momento em que um **Escravo** é criado pelo Driver Elipse SuperDriver, este **Escravo** entra em um estado de repouso até o momento em que um Tag entre no estado **Em Advise** e é colocado no *cache*. O Driver Elipse SuperDriver realiza um gerenciamento de todos os Tags declarados e que estejam no estado **Em Advise**. O algoritmo de gerenciamento dos Tags do Driver Elipse SuperDriver separa e adiciona Tags aos *caches* dos respectivos **Escravos**, à medida que estes Tags entram no estado **Em Advise**. A saída do estado **Em Advise** leva à retirada de um Tag do *cache*. Portanto, estar com o status **Em Advise** é condição para um Tag entrar em um *cache* de leitura e escrita de um **Escravo**.

A partir do momento que existe um ou mais Tags no *cache* de um **Escravo**, este **Escravo** imediatamente passa para um estado de leitura ou escrita e processa uma lista de Tags que precisam ser lidos ou escritos. Esta lista de Tags é processada conforme os passos a seguir:

1. Se há Tags de Comunicação:
2. O Driver do objeto **Escravo** é iniciado e conecta-se ao dispositivo remoto.
3. O Driver processa, um por um, os Tags de Comunicação.
4. O Driver processa, um por um, os Tags especiais que são guardados na memória do **Escravo**.
5. O Driver se desconecta e para.
6. Se há Tags especiais, estes Tags são atualizados com o conteúdo guardado na memória do **Escravo**.
7. Se há retentativas de serviços, retorna ao passo 1 (um).

Os Tags no estado **Em Advise** são sempre lidos ou escritos a cada expiração do tempo do modo **Varredura Geral**, e o período de varredura é definido pelo valor geral configurado na aba SuperDriver, ou seja, o gerenciamento de serviços do Driver Elipse SuperDriver não leva em consideração os valores de períodos de varredura, ou seja, a propriedade **Scan** de cada um dos Tags, a menos que o modo de varredura individual esteja habilitado para o respectivo Perfil destes Tags. Os Tags no estado **Em Advise** apenas são lidos ou escritos respeitando os *scans* individuais quando o modo **Varredura Individual** está habilitado.

Um **Escravo** volta ao estado de repouso enquanto aguarda até que o próximo Tag expire o tempo de *scan*. Acontecendo uma expiração de tempo, este **Escravo** monta uma nova lista de Tags a serem lidos ou escritos.

Este processo de execução descrito anteriormente é efetuado para qualquer **Escravo** subordinado ao Driver Elipse SuperDriver. Não é permitido que **Escravos** com portas TCP/IP iguais tenham as conexões realizadas ao mesmo tempo. Um **Escravo** que necessite de uma porta TCP/IP ocupada permanece em modo de espera até que este recurso seja desocupado.

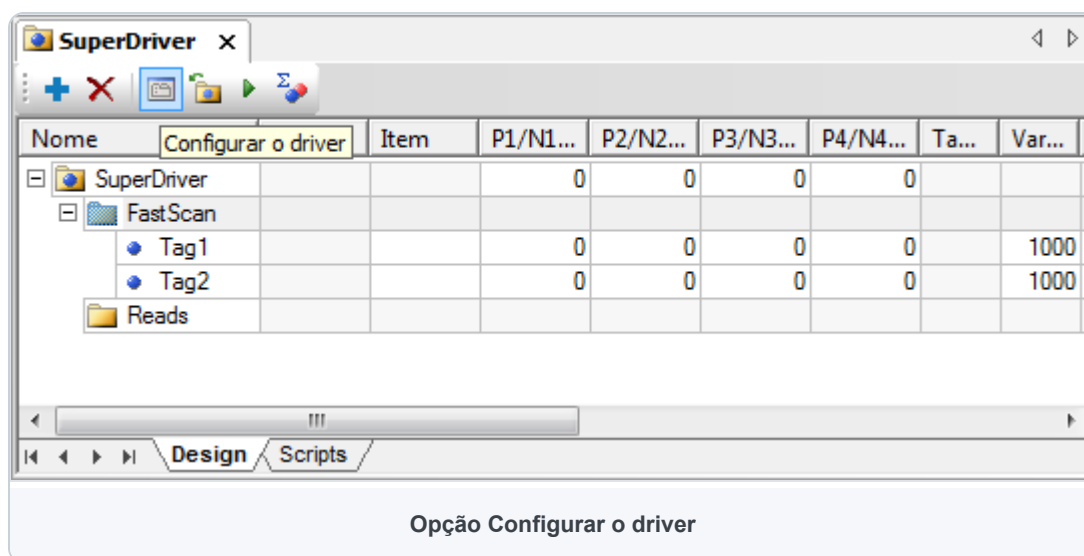
Comunicações Simultâneas

O Driver Elipse SuperDriver, quando em execução, gerencia a expiração dos tempos dos Tags **Em Advise** de todos os **Escravos**. Ao detectar a expiração dos tempos de um ou mais Tags, o Driver Elipse SuperDriver salva uma lista de **Escravos** com Tags de tempos expirados para terem leituras executadas tão logo seja possível. Esta lista pode ter o tamanho máximo de **Escravos** possíveis. No entanto, as tarefas de comunicação de todos os **Escravos** nem sempre podem ocorrer simultaneamente.

Toda tarefa de comunicação que ocorre em um **Escravo** consome recursos de processamento, ou uso de CPU, e ao menos um encadeamento de execução próprio (*thread*) é criado para cada comunicação. O sistema operacional limita o número máximo de *threads* por processo. Por esta razão, o Driver Elipse SuperDriver possui uma limitação máxima de comunicações ocorrendo ao mesmo tempo. Quando este limite é atingido, todos os **Escravos** que precisam comunicar permanecem em uma lista de espera, e entram em comunicação à medida que as comunicações atuais sejam terminadas.

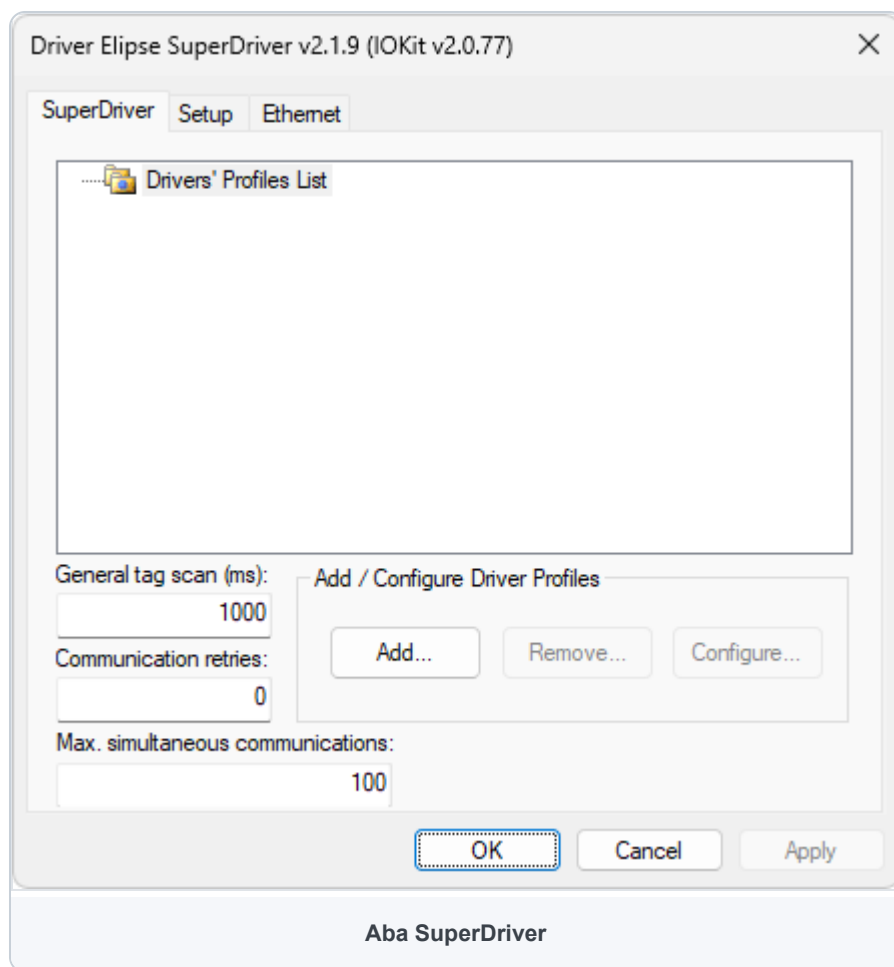
Configuração do Driver

O Driver Elipse SuperDriver não usa parâmetros **[P]** para configuração. Todas as configurações são realizadas na janela de propriedades, disponível na aba **Design** de um objeto Driver de Comunicação. Para isto, clique em **Configurar o Driver**, conforme a figura a seguir.



Aba SuperDriver

Ao clicar em **Configurar o Driver**, a janela da figura a seguir é aberta e a aba **SuperDriver** é selecionada.



As opções disponíveis nesta aba estão descritas na tabela a seguir.

Opções disponíveis na aba SuperDriver

OPÇÃO	DESCRIÇÃO
Drivers' Profiles List	Estrutura em árvore destinada à exibição da lista de Perfis de Drivers
General tag scan (ms)	Tempo, em milissegundos, do período de varredura, ou <i>scan</i> , relativo a todos os Tags que estão na lista de leitura e universal a todos os objetos Escravos . Isto significa que todos os Escravos procuram obedecer uma leitura cíclica de cada um dos Tags por este valor. A expiração do tempo de <i>scan</i> geral sempre gera um pedido de leitura de todos os Tags em lista de cada Escravo
Communication retries	Configura o número de retentativas de comunicação. Um objeto Escravo retenta a comunicação enquanto há Tags com leituras mal-sucedidas. Em cada modo de operação, o Driver Elipse SuperDriver age de maneira diferente com relação a este valor. Para mais informações, consulte o tópico Modos de Operação. O número máximo permitido de retentativas é 10

OPÇÃO	DESCRIÇÃO
Max. simultaneous communications	Configura o limite máximo de comunicações ocorrendo ao mesmo tempo. Para mais informações, consulte o tópico Comunicações Simultâneas. Quando este valor é atingido, todos os Escravos que precisam comunicar permanecem em uma lista de espera e entram em comunicação à medida que as comunicações atuais são terminadas. Este valor deve ser levado em conta pelas limitações físicas do processador do computador que executa o sistema. O valor mínimo configurável é 1 (um) e o valor máximo é 750. O melhor valor a ser configurado varia conforme a capacidade do processador do sistema em realizar tarefas em paralelo, portanto nem sempre o maior valor possível é o valor ótimo para o sistema, tal como em uma relação diretamente proporcional. O valor ótimo pode ser obtido por meio de testes com leituras simultâneas de um número de Escravos superior ao valor configurado
Add/Configure Driver Profiles	Permite adicionar, remover e configurar Perfis de Drivers. Para mais informações, consulte o tópico Configuração de Perfil

Clique em **OK** para confirmar as configurações listadas e fechar esta janela.

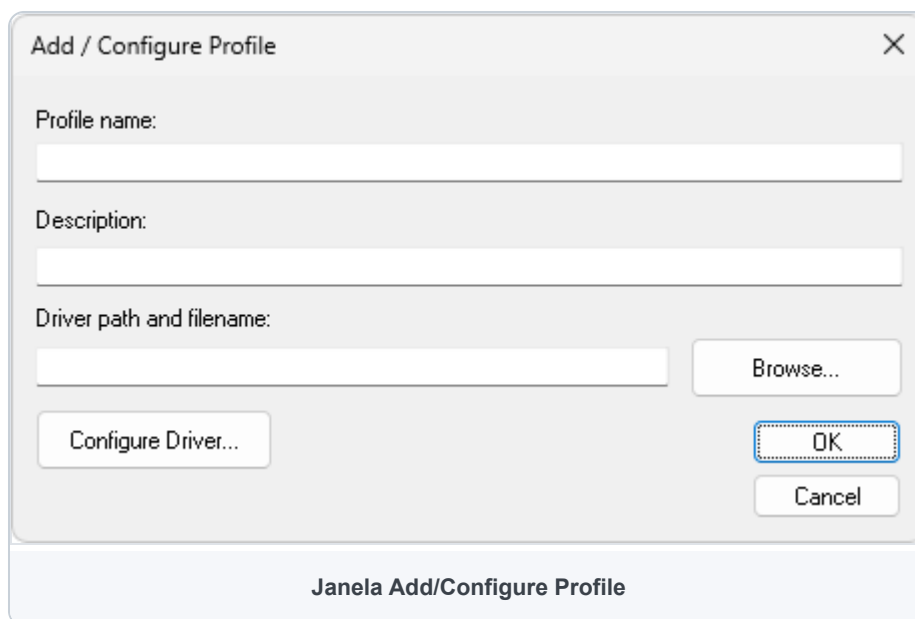
Abas Setup e Ethernet

As abas **Setup** e **Ethernet** são abas herdadas da biblioteca **IOKit** e são as únicas que devem ser levadas em conta. Consulte o tópico Documentação das Interfaces de Comunicação para mais informações sobre estas abas.

Diversos parâmetros configurados nestas abas são repassados aos Drivers **Escravos**. O tópico Parâmetros dos Perfis Herdados do SuperDriver especifica cada um destes parâmetros e deve ser consultado para mais informações sobre as configurações necessárias em cada Perfil.

Configuração de Perfil

Clique em **Add** ou **Configure** na aba SuperDriver para abrir a janela da figura a seguir e adicionar ou configurar um Perfil.



Dialog box titled "Add / Configure Profile" with a close button (X) in the top right corner. It contains three input fields: "Profile name:", "Description:", and "Driver path and filename:". Below the "Driver path and filename:" field is a "Browse..." button. At the bottom left is a "Configure Driver..." button. At the bottom right are "OK" and "Cancel" buttons. Below the dialog box, the text "Janela Add/Configure Profile" is displayed.

As opções disponíveis nesta janela estão descritas na tabela a seguir.

Opções disponíveis na janela Add/Configure Profile

OPÇÃO	DESCRIÇÃO
Profile name	Configura o nome de um Perfil. Selecione um nome que represente este Perfil de Driver ao referenciar Tags
Description	Configura a descrição de um Perfil com a finalidade de documentação. Apesar de não impactar no funcionamento de um Perfil, é uma opção obrigatória que pode ser preenchida com qualquer conteúdo com entrada por teclado
Driver path and filename	Caminho e nome de um arquivo de Driver selecionado para um Perfil. Clique em Browse para abrir a janela de seleção de arquivos padrão do Windows e selecionar um arquivo existente
Configure Driver	Abre a janela de configuração do Driver indicado na opção Driver path and filename . Como cada Driver tem particularidades, esta janela de configuração deve ser preenchida conforme o Manual do Usuário do respectivo Driver. Consulte o tópico Parâmetros dos Perfis Herdados do SuperDriver para evitar configurações desnecessárias

Parâmetros dos Perfis Herdados do SuperDriver

Os parâmetros a seguir são copiados da configuração do Driver Elipse SuperDriver para todos os Perfis subordinados, portanto não precisam ser preenchidos em todos os Perfis.

- **Aba Setup:**
- **Physical layer (Camada física):** Fixo em **Ethernet** e não prevê comunicação em outras interfaces

- **Connection management mode (Modo de gerenciamento de conexão):** Automático ou Manual
- **Retry failed connection every n seconds**
- **Give up after n failed retries**
- **Disconnect if non-responsive for n seconds**
- **Ativação e nome do arquivo de Log:** Gera um arquivo de log para cada **Escravo**, cujo nome de arquivo é composto pelo nome configurado no Driver Elipse SuperDriver com um sufixo composto pelo nome do Perfil e da Porta TCP/IP
- **Aba Ethernet:**
- **Transport (Protocolo de transporte):** Fixo em **TCP** e não prevê outro protocolo de transporte
- **Endereço IP principal:** Endereço IP de um servidor
- **Backup address:** Endereço IP de *backup*
- **Especificação de porta local (tanto para o IP principal quanto para o backup)**
- **Ping:** Habilitação, *time-out* e retentativas

NOTAS

- As demais configurações nas abas **Setup** e **Ethernet** do Driver Elipse SuperDriver são ignoradas e vale a configuração de cada Perfil.
- A porta TCP/IP da aba **Ethernet** é desconsiderada. Este parâmetro é especificado nos Tags.

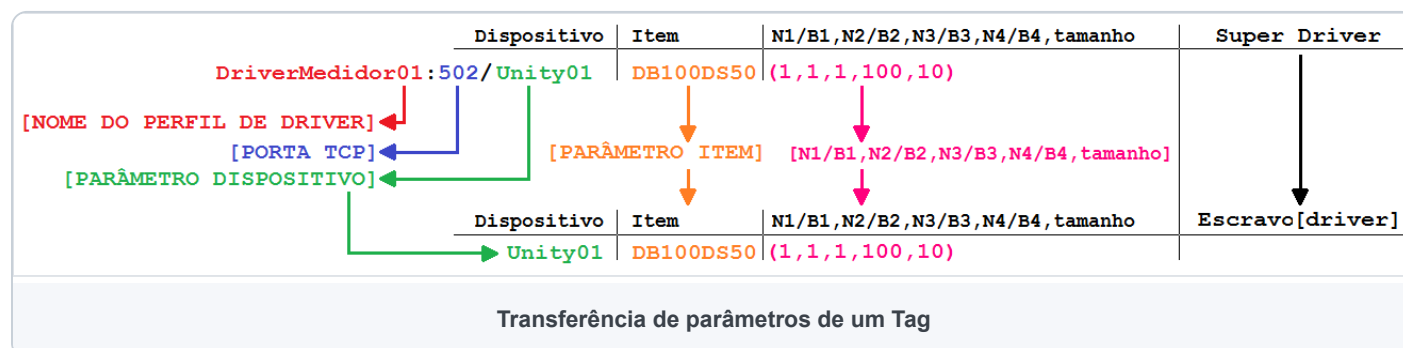
Referência de Tags

Esta seção contém informações sobre a configuração dos parâmetros *Dispositivo*, *Item* e *[N/B]* de cada Tag deste Driver. Os Tags são classificados nas categorias Tags de Comunicação e Tags Especiais.

Tags de Comunicação

Considera-se como **Tag de Comunicação** um Tag que é executado exclusivamente pela instância de Driver interna a um **Escravo**, através do repasse de parâmetros intermediado por este **Escravo**. O Driver Elipse SuperDriver cria um Tag virtual que é repassado para leitura ou escrita por um **Escravo**.

A figura a seguir esquematiza como é realizada a transferência de parâmetros de um Tag declarado no Driver Elipse SuperDriver para que um Tag virtual seja lido pelo Driver de um **Escravo**.



A documentação de ajuda de cada Driver utilizado como **Escravo** deve ser consultada para configuração dos parâmetros *N/B*, tamanho de Blocos e eventualmente os parâmetros *Dispositivo* e *Item*, caso estes sejam utilizados.

- **Dispositivo:** Para especificar um **Escravo** ao qual um Tag pertence, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]**. Caso o Driver **Escravo** utilize algum parâmetro *Dispositivo*, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/[PARÂMETRO DISPOSITIVO]**, obedecendo a documentação de ajuda
- **Item:** Caso o Driver **Escravo** utilize algum parâmetro *Item*, obedeça o que deve ser preenchido, segundo a documentação de ajuda
- **Parâmetros N/B e de tamanho:** Preencha conforme a documentação de ajuda

Os campos de valores de cada Elemento e de cada Tag PLC também têm as interpretações esclarecidas mediante a consulta da documentação de ajuda.

Tags Especiais

Considera-se um **Tag Especial** aquele Tag que é executado lendo apenas conteúdos memorizados por um **Escravo** e que podem realizar alguma função. Estes Tags, quando lidos, não dependem do carregamento, inicialização ou conexão do Driver de Comunicação de um **Escravo**, por isto têm um desempenho superior, e períodos de *scan* inferiores em relação à Tags de Comunicação podem ser exigidos. Os tipos de Tags especiais estão descritos nos tópicos a seguir.

IOLit

Para configurar um Tag da biblioteca **IOLit**, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/[TAG DO IOLIT]** no campo **Dispositivo** (*Device*), conforme o exemplo a seguir. Os demais campos são ignorados.

```
Profile02:502/IO.PhysicalLayerStatus [Tag PLC]
Profile02:502/IO.IOLitEvent [Tag Bloco com quatro Elementos]
```

Consulte o tópico Documentação das Interfaces de Comunicação para informações sobre os Tags da biblioteca **IOLit**.

IndividualScan

Este Tag, quando em modo **Advise**, indica a um **Escravo** subordinado ao Driver Elipse SuperDriver que deve realizar leituras dos Tags, respeitando os períodos de varredura individualmente configurados.

Para configurar um Tag **IndividualScan**, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/IndividualScan** no campo **Dispositivo**. Os parâmetros *[N]* são ignorados. A propriedade **Value** deste Tag mostra um valor numérico, incrementado a cada leitura realizada, conforme o exemplo a seguir.

```
Profile01:502/IndividualScan
```

CurrentMainIP

Este Tag indica o endereço IP ativo de uma conexão. Para configurá-lo, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/CurrentMainIP** no campo **Dispositivo** (*Device*). Os demais campos são ignorados. A propriedade **Value** deste Tag mostra uma **String** com o endereço IP ou *alias* do servidor ativo, conforme o exemplo a seguir.

`Profile01:502/CurrentMainIP`

ErrorHistory

Este Tag retorna uma lista de erros ocorridos durante a comunicação com o Driver de um **Escravo**. Quando não há erros, retorna uma lista vazia. Este Tag deve ser um Tag Bloco com até 2 (dois) Elementos. Para configurar este Tag, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/ErrorHistory** no campo **Dispositivo**. Os demais campos são ignorados, conforme o exemplo a seguir.

`Profile01:502/ErrorHistory`

A propriedade **Value** do Elemento 1 (um) contém um código numérico de erro. A interpretação do código obedece à tabela a seguir.

A propriedade **Value** do Elemento 2 (dois) contém uma **String** com o significado do erro, em inglês. Cada Driver possui um modo de exibir o significado de um erro. Portanto, para manter um padrão ou mesmo alterar o idioma, realize uma correspondência entre a tabela de códigos de erro e a aplicação.

Este Tag somente pode ser usado nos seguintes Drivers:

- Enron Modbus versão **1.2**
- Modicon Modbus versão **2.8**
- Elster-Instromet 999 versão **1**

Códigos de erro

CÓDIGO	CATEGORIA	DESCRIÇÃO
1	TIMEOUT	Um Driver não recebeu resposta de um equipamento no tempo de <i>time-out</i> configurado
2	RECEBIDO FRAME INVÁLIDO	Um Driver recebeu um <i>frame</i> inválido como resposta, como por exemplo um erro de checagem ou um campo de endereço diferente do solicitado
3	ARGUMENTO INVÁLIDO	Uma aplicação de usuário transferiu Tags com parâmetros inválidos para um Driver
4	ERRO DE CONEXÃO	Erro relacionado à camada física, como por exemplo uma falha na conexão TCP/IP ou porta serial inválida
5	SEM MEMÓRIA	O computador em que uma aplicação está em execução está sem memória, impedindo a alocação de recursos para as operações de um Driver
6	ERRO INTERNO	Este erro é retornado em casos inesperados, isto é, situações ou contextos detectados por um Driver que não podem ocorrer e

CÓDIGO	CATEGORIA	DESCRIÇÃO
		que podem ser considerados como falhas. Esta categoria de erro é mais rara e precisa ser encaminhada ao suporte técnico da Elipse Software
7	RECEBIDO FRAME DE EXCEÇÃO DO PROTOCOLO	Um <i>frame</i> de resposta válido foi recebido de um equipamento, porém informando uma condição de erro prevista por protocolo

Tags de Estatísticas e Bloqueio de Varreduras

Durante a execução, o Driver Elipse SuperDriver acumula contagens para estatísticas que podem ser acessadas por uma aplicação através de Tags específicos de Estatísticas de Coleta. Todos estes Tags são somente de leitura, com exceção do Tag **\$BlockScans**, que bloqueia todas as varreduras de todos os **Escravos** do Driver Elipse SuperDriver.

Para que um Tag seja reconhecido, o nome deve estar no campo **Dispositivo** (*Device*) e o campo **Item** deve estar em branco.

- **\$ScansActive**: Retorna a quantidade de varreduras em execução neste instante
- **\$ScansCompleted**: Retorna a quantidade de varreduras que um Driver já completou
- **\$IndividualScansActive**: Retorna a quantidade de varreduras individuais em execução neste instante
- **\$IndividualScansCompleted**: Retorna a quantidade de varreduras individuais que um Driver já completou
- **\$GeneralScansActive**: Retorna a quantidade de varreduras gerais em execução neste instante
- **\$GeneralScansCompleted**: Retorna a quantidade de varreduras gerais que um Driver já completou
- **\$BlockScans**: Permite bloquear a execução de scans. Os valores possíveis para este Tag são **0**: Bloqueia todas as varreduras ou **1**: Libera todas as varreduras. Este Tag também permite escritas

Parâmetros de Configuração

Somente Escrita

B1	-1 (menos um)
B2	0 (zero)
B3	0 (zero)
B4	3 (três)

Existe a possibilidade de alterar os parâmetros configurados na aba SuperDriver em tempo de execução, com este Driver em modo **Offline**. Todas as configurações *offline* utilizam a escrita de um Tag Bloco com 2 (dois) Elementos. Os tópicos a seguir mostram cada um dos Tags configuráveis desta forma. Para mais informações sobre o uso deste Tag, consulte o tópico Documentação das Interfaces de Comunicação.

GeneralTagScan

Utilize os Elementos a seguir para alterar o tempo do período de varredura geral de todos os Tags em modo **Advise**:

- **Elemento 1:** "SUPERDRIVER.GeneralTagScan"
- **Elemento 2:** Tempo do período de varredura, ou *scan*, em milissegundos

CommunicationRetries

Utilize os Elementos a seguir para alterar o número de retentativas de comunicação:

- **Elemento 1:** "SUPERDRIVER.CommunicationRetries"
- **Elemento 2:** Número de retentativas. O valor máximo permitido é 10

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **SuperDriver**.

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet
- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.1.10	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 38047</i>).
2.1.7	13/02/2017	M. Ludwig	• Corrigido um travamento na finalização em condições de estresse (<i>Case 21663</i>). • Corrigido um problema que disparava mais de uma coleta de forma simultânea no mesmo Perfil ou porta TCP/IP e poderia causar um GPF (<i>Case 21751</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.1.4	17/10/2016	M. Ludwig	- Corrigida a falta de carregamento das configurações em Perfis de Drivers para Escravos (Case 21664). - Driver portado para a biblioteca IOKit versão 2.0.72 (Case 21634).
2.1.2	17/03/2015	M. Ludwig	- Implementada a desistência das retentativas de comunicação caso se detecte um equipamento em modo Offline (Case 18356). - Implementada a sobrescrita das configurações Ethernet para evitar acionamento de redundância automática inadvertida em modo Varredura Geral (Case 18389).
2.1.1	14/11/2014	M. Ludwig	- Implementado um limite máximo de comunicações simultâneas (Case 15645). - Alterado o limite de Escravos para 1000. Ao atingir este limite, este Driver não cria mais Escravos (Case 17564).
2.0.5	21/03/2014	M. Ludwig	Adicionada uma limitação nas conexões de Escravos (Case 15976).
2.0.4	21/11/2013	F. Englert M. Ludwig	- Corrigida a possibilidade de travamento ao fechar este Driver (Case 14855). - Implementada uma segunda chamada de escritas após a execução de um processo de leitura (Case 15176).
2.0.3	27/09/2013	F. Englert	Adicionados Tags de estatísticas de coleta (Case 14954).
2.0.2	23/09/2013	F. Englert M. Ludwig	- Corrigidos vazamentos de memória (<i>heaps</i> do Windows) durante a execução (Case 14918). - Driver portado para a biblioteca IOKit versão 2 (Case 13888). - Implementada a geração de logs dos sub-Divers em arquivos diferentes (Case 13890). - O Driver Elipse SuperDriver agora só carrega Drivers que suportem múltiplas instâncias no mesmo processo (Case 13995).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.2.2	05/03/2013	M. Ludwig	• Alterado o comportamento de espera no encerramento (<i>Case 13139</i>). • Corrigida a atualização de parâmetros em tempo de execução (<i>Case 13358</i>). • Corrigida a falta de liberação de recursos nas escritas de Tags de comunicação (<i>Case 13484</i>). • Corrigido o crescimento indefinido de uso de recursos quando o Tag ErrorHistory não é lido (<i>Case 13477</i>). • Corrigido o carregamento incorreto de endereços IP configurados na janela de propriedades (<i>Case 13476</i>).
1.1.1	31/05/2012	C. Mello	• Adicionado um serviço de escrita de Tags para Perfis de Driver com Tags em modo Advise (<i>Case 12521</i>).
1.0.1	22/09/2011	M. Ludwig	• Versão inicial deste Driver (<i>Case 11581</i>).