



ABDF1

v2.0.18 · Português

Conteúdo

Driver Allen Bradley DF1

Preparando o Equipamento

Parâmetros [P] de Configuração do Driver

Janela de Configuração do Driver

Janela de Configurações Avançadas

Referência de Tags

Parâmetros [N] de Endereçamento para Tags do Tipo PLC

Parâmetros [B] de Endereçamento para Tags do Tipo Bloco

Números de Arquivos

Tipos de Memória

Exemplos de Endereçamento

Configuração de Tags Através de Inteiros e Strings

Solução de Problemas Comuns

Leitura com Superblocos

Modo Legado

Documentação das Interfaces de Comunicação

Histórico de Revisões do Driver

Driver Allen Bradley DF1

Nome do Arquivo	ABDF1.dll
Fabricante	Allen-Bradley (Rockwell Automation)
Equipamentos	PLC5, SLC500, Micrologix e ControlLogix
Protocolo	DF1, DF1 em CSPv4, PCCC e Ethernet/IP
Versão	2.0.18
Última Atualização	22/08/2025
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Sim
Nível	0

Introdução

O Driver Allen Bradley DF1 permite a comunicação serial entre sistemas da **Elipse Software** e os controladores programáveis da Allen-Bradley (Rockwell Automation) compatíveis com o protocolo DF1 (Serial) e via Ethernet, diretamente no canal Ethernet para o modelo SLC5/05 e via módulo 1761-NET-ENI para os modelos PLC5, Micrologix e SLC5/0X.

Para os equipamentos ControlLogix deve ser utilizado preferencialmente o Driver Allen-Bradley Ethernet/IP (ABCIP), mas também é possível utilizar este Driver com um mapeamento das variáveis internas dos equipamentos ControlLogix para o formato **DF1**.

Preparando o Equipamento

Esta seção contém informações sobre a configuração dos parâmetros **[P]** deste Driver.

Parâmetros **[P]** de Configuração do Driver

P1	Não utilizado
P2	Não utilizado
P3	Não utilizado
P4	Não utilizado

O Driver Allen Bradley DF1 utiliza a biblioteca **IOKit**, permitindo a comunicação via RS232 (DF1), Ethernet (DF1), Ethernet CSP, Ethernet/IP e Ethernet/IP com mensagens PCCC. Todos os parâmetros são definidos nas janelas de configuração da biblioteca **IOKit**. Para mais informações sobre esta biblioteca, consulte o tópico Documentação das Interfaces de Comunicação.

Janela de Configuração do Driver

Através da janela de configurações deste Driver, mostrada na figura a seguir, pode-se informar as configurações gerais.

The image shows a software configuration window titled "Driver Allen Bradley DF1 v2.0.17 (IOKit v2.0.137)". It features a tabbed interface with the following tabs: "Rockwell AB DF1" (selected), "Setup", "Serial", "Ethernet", "Modem", and "RAS". The "Rockwell AB DF1" tab contains the following settings:

- Default slave address: 0
- Master Address: 0
- PLC Type: SLC500 (dropdown menu)
- Check Type: BCC (dropdown menu)
- Transport: DF1 (dropdown menu)
- Mode: Full-Duplex (dropdown menu)
- Max Message Bytes: 240
- ☐ Use hexadecimal addressing on n1/b1
- Advanced Settings... (button)

At the bottom of the window are three buttons: "OK", "Cancel", and "Apply". Below the window, the text "Aba Rockwell AB DF1" is displayed.

Na aba **Rockwell AB DF1** encontram-se as configurações específicas deste Driver. Para informações sobre as demais abas, consulte o tópico Documentação das Interfaces de Comunicação. A tabela a seguir descreve as opções de configuração desta aba.

Opções disponíveis na aba Rockwell AB DF1

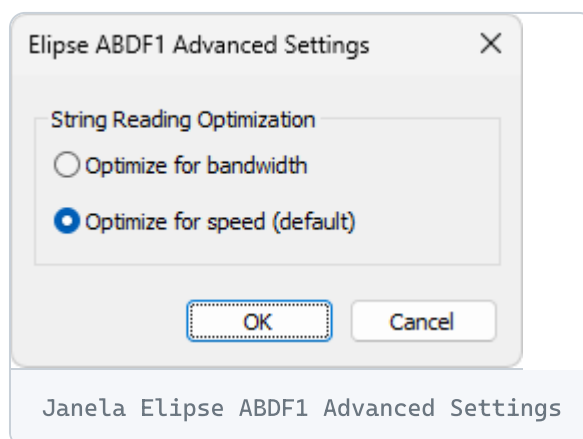
OPÇÃO	DESCRIÇÃO
Default slave address	Endereço padrão de um CLP. Utilizado sempre que no endereçamento dos Tags não é informado o endereço. Para a comunicação Ethernet CSP e Ethernet/IP (ENIP) , o endereço é sempre 0 (zero)
Master Address	Endereço padrão de um Mestre . Para comunicação Ethernet , o endereço é 0 (zero)
PLC Type	Tipo de CLP. As opções disponíveis são SLC500 , PLC5 , MicroLogix 1100 ou MicroLogix 1500 (porta Ethernet local) e MicroLogix ou ControlLogix (através de conversor Ethernet, emulando DF1)
Check Type	Tipo de verificação. As opções disponíveis são BCC ou CRC (somente para DF1)
Transport	Modo de transporte. As opções disponíveis são DF1 (geralmente se a comunicação é serial), CSPv4 (Ethernet) para SLC5/05 e ENIP se conectado através do conversor 1761 NET-ENI ou pela porta Ethernet do Micrologix 1100
Mode	Informa se é usado o modo Half-duplex ou o modo Full-duplex . Esta opção só é relevante se é usado o formato DF1 como camada de transporte
Max Message Bytes	Este parâmetro define o número máximo de bytes da área de dados de cada <i>frame</i> de comunicação. Em caso de blocos com mais bytes, este Driver divide a comunicação em mais <i>frames</i> , cada um contendo no máximo o limite definido nesta opção. O valor padrão desta opção é 240. Verifique na documentação do equipamento se este valor pode ser aumentado. Em caso de dúvida, use o valor padrão

OPÇÃO	DESCRIÇÃO
Use hexadecimal addressing on n1/b1	Se esta opção está habilitada, os parâmetros <i>N1</i> e <i>B1</i> dos Tags são alterados usando um endereçamento em formato hexadecimal. O byte mais significativo (MSB) indica o CLP de destino, enquanto o byte menos significativo (LSB) indica o número do arquivo. Com esta opção habilitada, é possível endereçar até 256 CLPs, o que não era possível com o endereçamento usado em versões anteriores, uma vez que o valor máximo dos parâmetros <i>N</i> e <i>B</i> é 65535 (16 bits). Com a opção habilitada, deve-se fornecer o valor dos parâmetros <i>N1</i> ou <i>B1</i> sempre em formato hexadecimal, adicionando o caractere "h" no Elipse SCADA ou o prefixo "&H" no E3 , no Power ou no Water . Para mais informações, verifique a Referência da Linguagem VBScript. Deixe esta opção desabilitada caso seja necessário manter a compatibilidade com versões anteriores deste Driver
Advanced Settings	Abre a Janela de Configurações Avançadas deste Driver, descrita no tópico a seguir

O restante dos parâmetros, como já mencionado, deve ser informado nas abas da biblioteca **IOKit**. Para a comunicação serial, o padrão é **19200 bps**, **8 databits**, **1 stopbit**, **no parity** e **check type** igual a **CRC**.

Janela de Configurações Avançadas

A janela de configurações avançadas deste Driver, mostrada na figura a seguir, é aberta clicando-se em **Advanced Settings** na aba **Rockwell AB DF1** da Janela de Configurações deste Driver.



Esta janela reúne as configurações menos usadas, que em geral não requerem a atenção do usuário e podem ser mantidas nas opções padrão. Atualmente, as seguintes configurações avançadas estão disponíveis:

- **String Reading Optimization**
- **Optimize for bandwidth:** A cada leitura de **Strings**, este Driver executa 2 (duas) requisições a um equipamento. Na primeira requisição é lido apenas o valor numérico que indica o tamanho real da **String**, e na segunda requisição a **String** é lida com o tamanho detectado. Este procedimento tende a economizar o uso de banda, pois é lido apenas o tamanho exato da **String**. Este era o comportamento padrão deste Driver até a versão **2.0.3**
- **Optimize for speed (default):** Esta é a opção padrão para novas instâncias deste Driver, criadas com bibliotecas da versão **2.0.3** ou mais recentes. Nesta opção, este Driver executa uma única requisição para cada leitura de **Strings**, lendo sempre o tamanho máximo permitido, 82 caracteres, e descartando posteriormente os bytes não utilizados. Tende a deixar a leitura de **Strings** com aproximadamente o dobro da velocidade em comparação com a opção anterior, mas pode exigir a leitura de mais caracteres nesta requisição única, sempre 82 caracteres, do que seriam necessários

NOTA

Ao atualizar a biblioteca deste Driver em instâncias inicialmente criadas com versões anteriores à versão **2.0.3** em aplicações antigas, mantém-se o comportamento legado, ou seja, com a opção **Optimize for bandwidth** selecionada.

Referência de Tags

Esta seção contém informações sobre a configuração dos parâmetros **[N/B]** deste Driver.

Parâmetros **[N]** de Endereçamento

para Tags do Tipo PLC

N1	Endereço de um CLP (<i>PC source</i>) $\times 1000$ + número de um arquivo. Se um endereçamento hexadecimal é usado, MSB é igual ao endereço de um CLP e LSB é igual ao número de um arquivo. Para mais informações, consulte a tabela Números de Arquivos
N2	Tipo de variável. Para mais informações, consulte a tabela Tipos de Memória
N3	Número do elemento inicial de um arquivo
N4	Número do subelemento inicial de um arquivo. Para mais informações, consulte a nota a seguir

Se o endereço padrão de um CLP é configurado, o parâmetro *N1* é somente o número de um arquivo.

NOTA

Para manipulação de dados com arquivos do tipo **ASCII**, ou seja, com o parâmetro *N2* igual a 31, o parâmetro *N4* é utilizado para definir o tamanho dos dados (**String**), pois este tipo de arquivo não utiliza endereçamento com subelementos. Como a alocação interna dos dados ocupa um registrador de 16 bits, somente comprimentos de **String** com tamanho par de caracteres são permitidos.

Parâmetros [B] de Endereçamento

para Tags do Tipo Bloco

B1	Endereço de um CLP (<i>PC source</i>) $\times 1000$ + número de um arquivo. Se o endereçamento é em hexadecimal, MSB é igual ao endereço de um CLP e LSB é igual ao número de um arquivo. Para mais informações, consulte a tabela Números de Arquivos
B2	Tipo de variável. Para mais informações, consulte a tabela Tipos de Memórias
B3	Número do elemento inicial de um arquivo
B4	Número do subelemento inicial de um arquivo. Para mais informações, consulte a nota a seguir
Size	Número de Elementos de Bloco

Se o endereço padrão de um CLP é configurado, o parâmetro *B1* é somente o número de um arquivo.

NOTA

Para manipulação de dados com arquivos do tipo **ASCII**, ou seja, com o parâmetro *B2* igual a 31, o parâmetro *B4* é utilizado para definir o tamanho dos dados (**String**), pois este tipo de arquivo não utiliza endereçamento com subelementos. Como a alocação interna dos dados ocupa um registrador de 16 bits, somente comprimentos de **String** com tamanho par de caracteres são permitidos.

Números de Arquivos

Números de Arquivos

ARQUIVO	NÚMERO (N1 OU B1)	TIPO	DESCRIÇÃO
O0	0 (zero)	Output	Status dos terminais de saída
I1	1 (um)	Input	Status dos terminais de entrada
S2	2 (dois)	Status	Informações de operação de um CLP
B3	3 (três)	Bit	<i>Internal relay logic</i>
T4	4 (quatro)	Timer	Temporizador (<i>timer</i>), valores e bits de status pré-definidos
C5	5 (cinco)	Counter	Contador (<i>counter</i>), valores e bits de status pré-definidos
R6	6 (seis)	Control	Tamanho, posição do ponteiro e bits de status para instruções específicas como deslocamento (<i>shift</i>) de registradores
N7	7 (sete)	Integer	Armazena valores numéricos e informações de bit
F8	8 (oito)	Float	Armazena um número com um intervalo entre 1,1754944E-38 e 3,40282347E+38
F8	8 (oito)	Definido pelo usuário	Arquivos definidos pelo usuário para armazenamento de valores Bit , Timer , Counter , Control ou Integer
P9	9 (nove)	PID	Arquivos PID que contém diversas variáveis de controle
10-255	10 a 255	Definido pelo usuário	Arquivos definidos pelo usuário para armazenamento de valores Bit , Timer , Counter , Control ou Integer
10-255	10 a 255	Definido pelo usuário	Arquivos definidos pelo usuário para armazenamento de valores String (ST) ou ASCII (A) quaisquer

NOTA

Os arquivos podem variar de acordo com o modelo de CLP.

Tipos de Memória

Tipos de Memória (variável)

VARIÁVEL	ARQUIVO	TIPO (N2 OU B2)
Word	SLC_OUTPUT	0 (zero)
	SLC_INPUT	1 (um)
	SLC_STATUS	2 (dois)
	SLC_BIT	3 (três)
	SLC_TIMER	4 (quatro)
	SLC_COUNTER	5 (cinco)
	SLC_CONTROL	6 (seis)
	SLC_INTEGER	7 (sete)
	SLC_INTEGER, um inteiro de 16 bits com sinal	71. Para mais informações, consulte a nota a seguir
BCD	SLC_OUTPUT	8 (oito)
	SLC_INPUT	9 (nove)
	SLC_STATUS	10
	SLC_BIT	11
	SLC_TIMER	12
	SLC_COUNTER	13
	SLC_CONTROL	14
	SLC_INTEGER	15
Bit	SLC_OUTPUT	16, apenas Tags individuais
	SLC_INPUT	17, apenas Tags individuais

VARIÁVEL	ARQUIVO	TIPO (N2 OU B2)
	SLC_STATUS	18, apenas Tags individuais
	SLC_BIT	19, apenas Tags individuais
	SLC_TIMER	20, apenas Tags individuais
	SLC_COUNTER	21, apenas Tags individuais
	SLC_CONTROL	22, apenas Tags individuais
	SLC_INTEGER	23, apenas Tags individuais
Float	SLC_FLOAT	24
Long	SLC_LONG	25
String	SLC_STRING	26, apenas Tags individuais
Bit	SLC_LONG	28
Float	SLC_INTEGER	29
Word	SLC_PID. Para mais informações, consulte a nota a seguir	30
ASCII	SLC_ASCII	31

NOTAS

- Os tipos de memória 7 (sete) e 71 acessam o mesmo dado em um CLP. A diferença entre ambos é apenas a interpretação dos dados, que no primeiro caso são interpretados como inteiros de 16 bits sem sinal (**Word**) e no segundo caso como inteiros de 16 bits com sinal.
- Blocos para leitura de arquivos PID devem conter no máximo 23 Elementos, endereçados de 0 (zero) até 22.

Exemplos de Endereçamento

Exemplos de Endereçamento no Formato N1.N2.N3.N4

- Usando o endereçamento decimal:

```
CLP = 10, F8:40 (10008.24.40.0) (File 8, SLC_FLOAT)
CLP = 2, N25:100 (2025.7.100.0) (File 25, SLC_INTEGER)
CLP = 1, C5:42.ACC (1005.5.42.2) (File 5, SLC_COUNTER, Elemento 3)
CLP = 3, C5:42.PRE (3005.5.42.1) (File 5, SLC_COUNTER, Elemento 2)
```

- Usando o endereçamento hexadecimal:

```
CLP = 10, F8:40 (0A08h.24.40.0) (File 8, SLC_FLOAT)
CLP = 2, N25:100 (0219h.7.100.0) (File 25, SLC_INTEGER)
CLP = 1, C5:42.ACC (0105h.5.42.2) (File 5, SLC_COUNTER, Elemento 3)
CLP = 3, C5:42.PRE (0305h.5.42.1) (File 5, SLC_COUNTER, Elemento 2)
```

Exemplos de Endereçamento para Arquivos no Formato ASCII (N2 ou B2 igual a 31)

Arquivos no formato **ASCII** são endereçados por elementos de 16 bits e os dados são manipulados com base em um comprimento n de dados, todos em sequência, representando uma cadeia de caracteres. Por exemplo, para receber 20 caracteres a partir do endereço 10 de um arquivo no formato **ASCII**, configure um Tag PLC da forma indicada a seguir.

```
CLP = 1, A9:10 (1009.31.10.20) (File 9, SLC_ASCII, 20 caracteres a partir do Elemento 10 do arquivo)
```

No caso de Tags Bloco, o comprimento n de dados é processado para cada um dos Elementos. Por exemplo, para receber 20 caracteres em cada um dos três Elementos de um Tag Bloco, a partir do endereço 10 de um arquivo no formato **ASCII**, configure um Tag Bloco da forma indicada a seguir.

CLP = 1, A9:10 (1009.31.10.20) (File 9, SLC_ASCII, 20 caracteres a partir do Elemento 10 do arquivo, para cada um dos Elementos do Tag Bloco)

Tag.Elemento1 = 20 caracteres a partir do Elemento 10 do arquivo

Tag.Elemento2 = 20 caracteres a partir do Elemento 20 do arquivo

Tag.Elemento3 = 20 caracteres a partir do Elemento 30 do arquivo

NOTAS

- Como a alocação interna dos dados ocupa um registrador de 16 bits, a configuração no parâmetro *N4* ou *B4* permite somente comprimentos de **String** com um tamanho par de caracteres.
- As leituras de **Strings** estão condicionadas ao comprimento configurado no parâmetro *N4* ou *B4*, mas pode retornar um tamanho inferior se um caractere **Null** (0x00), ou terminador de **String**, é recebido.
- As escritas de **Strings** com comprimento superior ao configurado no parâmetro *N4* ou *B4* têm os caracteres truncados até o limite deste parâmetro.
- As escritas de **Strings** com comprimento inferior ao configurado no parâmetro *N4* ou *B4* são preenchidas com caracteres **Null** (0x00) à direita até completar o comprimento definido neste parâmetro.

Configuração de Tags Através de Inteiros e Strings

Este Driver permite a configuração de Tags através de **Inteiros** e **Strings**. Para isto, devem ser usadas as propriedades **Dispositivo** (**ParamDevice**) e **Item** (**ParamItem**), respectivamente.

A propriedade **Dispositivo** dos Tags tem o valor herdado deste Driver, pode ser sobrescrita em um Tag e define o endereço de um equipamento acessado por este Tag. Caso esta propriedade seja deixada vazia, automaticamente assume-se o valor 0 (zero).

Já a propriedade **Item** identifica um dado acessado por um Tag em um equipamento. Esta propriedade pode ter um dos formatos a seguir, em que os valores entre colchetes são opcionais.


```
[TIPODADO] AREA ARQUIVO [: ENDereco] / [BIT]
```

Ou para *timers* e *counters*:

```
[TIPODADO] AREA ARQUIVO : ENDereco. SUBELEMENTO
```

Ou para arquivos no formato **ASCII**:

```
[TIPODADO] AREA ARQUIVO : ENDereco - TAMANHO DO DADO (STRING)
```

As tabelas a seguir definem os mnemônicos que podem ser usados nos valores descritos anteriormente.

Tipos de variáveis

TIPO DA VARIÁVEL	STRING	PARÂMETRO N2 OU B2
Word	W	Entre 0 (zero) e 7 (sete)
Long	DW, usado apenas com a área L	25
Signed Word	SW, usado somente com tipos de dados inteiros I	71
BCD	BCD	Entre 8 (oito) e 15
Bit	Sempre que o valor opcional Bit é definido	Entre 16 e 23
Float	Não definido. Para usar este tipo de variável, defina a área como F	24
String	Não definido. Para usar este tipo de variável, defina a área como ST	26
ASCII	Não definido. Para usar este tipo de variável, defina a área como A	31

Áreas

ÁREA	STRING
SLC_OUTPUT	O
SLC_INPUT	I
SLC_STATUS	S
SLC_BIT	B
SLC_TIMER	T
SLC_COUNTER	C
SLC_CONTROL	R
SLC_INTEGER	N
SLC_FLOAT	F
SLC_LONG	L, sempre com um tipo de variável DW
SLC_STRING	ST
SLC_ASCII	A
SLC_PID	PD, sempre com um tipo de variável W

A tabela a seguir contém exemplos de Tags configurados por **Strings**, relacionando-os com os parâmetros *N* ou *B* equivalentes.

Exemplos de Tags definidos por Strings

DISPOSITIVO	ITEM	N1 OU B1	N1 OU B1 (HEX)	N2 OU B2	N3 OU B3	N4 OU B4
2 (dois)	WO0:0/4	2000	0200h	16	0 (zero)	4 (quatro)
8 (oito)	SWI1:86	8001	0801h	71	86	0 (zero)
8 (oito)	WS2:32	8002	0802h	2 (dois)	32	0 (zero)
0 (zero)	WI1:104	1 (um)	1 (um)	1 (um)	104	0 (zero)
11	BCDT4:38.9	11004	0B04h	12	38	9 (nove)
10	F8:3	10008	0A08h	24	3 (três)	0 (zero)
6 (seis)	ST4:30	6004	0604h	26	30	0 (zero)
13	L7:90	13007	0D07h	25	90	0 (zero)
13	DWL7:100	13007	0D07h	25	100	0 (zero)
11	PD4:5.15	11004	0B04h	30	5 (cinco)	15
3 (três)	A9:10-20	3009	0309h	31	10	20

Solução de Problemas Comuns

1. Ao tentar comunicar com um equipamento, deve-se ter certeza que este equipamento não está comunicando com outra aplicação. Normalmente ocorre de o usuário realizar algumas configurações usando o RS Linx e não desativar a comunicação deste com um CLP ao encerrar a comunicação.
2. No caso de tentativas de leitura ou escrita de um elemento inexistente, o sistema indica um erro de comunicação. Isto também acontece caso se tente ler um Bloco com mais **Words** do que o arquivo especificado possui.
3. Só é possível configurar um Tag para comunicar com um bit se o subelemento de um arquivo é igual a 0 (zero), pois ao configurar o parâmetro *N2* ou *B2* para acessar um bit, este Driver considera que o subelemento é igual a 0 (zero). Caso haja necessidade de buscar estes bits e o

subelemento seja diferente de 0 (zero), aconselha-se configurar um Tag para buscar um dado completo e acessar estes bits com as funções existentes no **E3**, no **Power**, no **Water** ou no **Elipse SCADA**.

4. Ao tentar ler e escrever diretamente em entradas e saídas, ou seja, em arquivos **O0** e **I1**, dependendo do modelo e versão de um equipamento, podem ocorrer problemas de formatação numérica no endereçamento, gerando erros de comunicação registrados no log com a mensagem "AB_command returned error in status = 10!". Sugere-se o uso da técnica de mapeamento das entradas e saídas para inteiros, descrita no artigo Problemas de endereçamento ao acessar elementos de arquivos de entradas e saídas (O0 e I1) com o Driver Allen-Bradley ABDF1 do **Elipse Knowledgebase**.
5. Ao tentar comunicar com um equipamento MicroLogix 1100 através da porta Ethernet local e ocorrerem falhas, mude a configuração **PLC Type** para **MicroLogix 1500**. Algumas versões de *firmware* deste equipamento requerem este ajuste neste Driver.

Leitura com Superblocos

Este Driver tem suporte a Superblocos desde a versão **1.7**. Este recurso permite ao **E3**, ao **Power** ou ao **Water** agrupar os Tags configurados em uma aplicação em blocos com o maior tamanho possível, a fim de otimizar a comunicação. Para habilitar este recurso no **E3**, no **Power** ou no **Water**, configure a propriedade **EnableReadGrouping** de um objeto Driver com o valor Verdadeiro.

Nem todos os tipos de dados possuem suporte a este recurso. Os tipos de dados **BCD** e as áreas **Counters** e **Timers** não são agrupáveis.

O suporte ao agrupamento de blocos de bits ainda não está implementado na versão atual deste Driver, mas deve ser implementado em futuras versões.

O tipo de dados **29**, que lê inteiros de 16 bits e os formata como **Floats** de 32 bits, também não é agrupável na versão atual.

Para os Tags com saídas no formato **String**, recomenda-se a utilização de tabelas **A**, ou **ASCII**, para agrupamento das leituras de **Strings** via Superblocos. Caso um projeto opte pela utilização de tabelas **ST**, ou **String**, não é possível aplicar a leitura em agrupamento para este tipo de tabela.

Modo Legado

Para manter a compatibilidade com aplicações antigas, na versão **1.15** deste Driver foi criado um modo de operação chamado de **Modo Legado**. Neste modo, este Driver se comporta da mesma forma que na versão **1.14**. Este modo pode ser habilitado ou desabilitado das seguintes formas:

- Ao incluir um novo Driver em uma aplicação e carregar o arquivo ABDF1.dll, este modo está desabilitado
- Se já existe uma versão anterior deste Driver em uma aplicação e o arquivo ABDF1.dll é substituído pela versão mais recente, este modo está habilitado

Além disto, existe a opção de habilitar ou desabilitar em tempo de execução através do Tag **Set Configuration Parameters** da biblioteca **IOKit**. O parâmetro a ser configurado é *ABDF1.LegacyMode*, e o valor deve ser 0 (zero) para desabilitar e 1 (um) para habilitar.

Para mais informações sobre a utilização do Tag **Set Configuration Parameters** da biblioteca **IOKit**, consulte o tópico Documentação das Interfaces de Comunicação - Configurações Gerais - Tags de Comunicação.

Para mais informações sobre as diferenças entre as versões deste Driver, consulte o Histórico de Revisões do Driver.

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **ABDF1**.

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
2.0.18	22/08/2025	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 37930</i>).
2.0.17	08/12/2023	P. H. Santos	• Implementada a validação de Tags e assumido o valor 0 (zero) para a propriedade Dispositivo se esta propriedade está vazia (<i>Case 35216</i>).
2.0.16	19/08/2019	C. Mello	• Atualização de plataforma no código fonte deste Driver (<i>Case 27354</i>).
2.0.15	07/03/2019	C. Mello	• Adicionado suporte para leitura em Superblocos para arquivos em formato ASCII (<i>Case 24440</i>).
2.0.14	28/01/2019	C. Mello	• Adicionado suporte para uso de Tags Bloco em arquivos no formato ASCII (<i>Case 26033</i>).
2.0.13	17/09/2018	C. Mello	• Adicionado suporte para manipulação de dados em arquivos no formato ASCII (<i>Case 25046</i>).
2.0.10	13/02/2017	A. Hertzog	• Adicionado suporte a arquivos PID (<i>Case 21560</i>).
2.0.6	15/04/2016	M. Salvador	• Implementado suporte a escritas e leituras diretas de bits em áreas SLC_BIT (<i>Case 20093</i>).
2.0.5	10/09/2015	A. Quites	• Adicionada uma opção avançada de otimização para a leitura de Strings (<i>Case 18436</i>).
2.0.1	26/09/2013	G. Taschetto	• Driver portado para a biblioteca IOKit versão 2 (<i>Case 14126</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.15.1	21/11/2012	A. Quites G. Taschetto	• Implementado um controle de transação do protocolo DF1 encapsulado em ENIP, permitindo a rejeição de quaisquer <i>frames</i> de resposta atrasados (<i>Case 13090</i>). • O agrupamento por Superblocos foi desabilitado para Tags de acesso a arquivos de entrada e saída (parâmetro <i>N1</i> ou <i>B1</i> igual a zero ou igual a um), para evitar problemas com o eventual agrupamento de cartões não adjacentes (<i>Case 13084</i>). • Adicionada uma sugestão de mapeamento de entradas e saídas digitais para variáveis inteiras no tópico Solução de Problemas Comuns (<i>Case 12243</i>). • Corrigido um problema em que não era possível escrever em bits superiores ao bit 15 em cartões de I/O de 32 bits ou mais (<i>Case 13274</i>). • Adicionada uma proteção na configuração de Tags por Strings contra o uso de bits no tipo de dados Float e de subelementos nos tipos de dados Integer e Float (<i>Case 13339</i>). • Adicionado suporte a cartões de I/O de até 64 bits (<i>Case 13330</i>).
1.14.1	01/12/2010	A. Quites	• Corrigido um erro na escrita de bits de variáveis do tipo de dados Long (<i>Case 11923</i>).
1.13.1	13/10/2010	A. Quites	• Corrigido um erro na escrita de bits do tipo de dados Word mais significativo de tipos de dados de 32 bits (<i>Case 10768</i>). • Adicionada uma verificação de transação (TNS) da camada DF1 quando encapsulada em CSPv4 (<i>Case 11023</i>). • Corrigido um erro na escrita de Elementos de Bloco do tipo de dados Float (<i>Case 11024</i>). • Driver portado para o Windows CE (<i>Case 10913</i>). • Corrigido um erro na leitura do tipo de dados 71 (I16) no E3, com Superblocos (propriedade EnableReadGrouping) desabilitado (<i>Case 11653</i>). • Corrigidos os exemplos de endereçamento do subelemento ACC (<i>Case 10979</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.12.1	22/04/2009	A. Quites	• Corrigido um erro na leitura de Tags internos da biblioteca IOKit (Case 10242). • Corrigido um erro de deslocamento (<i>offset</i>) ao escrever em Elementos de Bloco com tipos de dados de 32 bits (Case 10253). • Corrigido um erro ao escrever nos bits de 16 a 31 em tipos de dados de 32 bits (Case 10345). • Implementado um novo tipo de dados 29, que lê dois Words e os converte para Float (Case 10016). • Corrigido um erro de deslocamento (<i>offset</i>) ao ler Blocos de tipos de dados de 32 bits com tamanho total superior ao limite máximo configurado (Case 10296). • Suporte a Superblocos de tipos de dados de 32 bits desabilitado, devido a um erro com o endereçamento de 16 bits (Case 10297).
1.11.1	07/10/2008	M. Salvador A. Quites	• Adicionado suporte a equipamentos Micrologix 1500 série B (Case 9132). • Adicionado suporte a escrita de bits em tipos de dados Long (Case 9743). • Adicionado um tipo de dados inteiro com sinal, correspondente ao parâmetro <i>N2</i> igual a 71 (Case 9800).
1.10.1	28/02/2007	A. Quites	• Resolvido um erro de TNS síncrono (Case 7833).
1.9.1	10/10/2006	M. Salvador	• Adicionado suporte a equipamentos Micrologix 1100 (Case 7454).
1.8.1	28/08/2006	A. Quites	• Corrigido um erro na recepção de pacotes em BCC (Case 7289).
1.7.1	26/07/2006	A. Quites M. Salvador	• Implementado o recurso de Superblocos e configuração de Tags por Strings (Case 6742). • Corrigido um erro na leitura e escrita de bits de entrada e saída (Case 6415). • Corrigido um problema na recepção nos modos Half-Duplex e Full-Duplex.

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.6.1	18/04/2004	M. Salvador	• Implementado um controle do número máximo de bytes na recepção. • Implementada a leitura de blocos maiores que o número máximo de bytes. • Implementada a escrita de bits.
1.5.1 Beta	15/04/2004	A. Quites M. Salvador	• Implementado o modo Half-Duplex na camada DF1 (<i>Case 5160</i>). • Criada a opção de configuração Use hexadecimal addressing on N1/B1.
1.2.1	24/06/2004	M. Salvador	• Versão anterior ao controle de revisões.
1.0.1	31/05/2004	M. Salvador	• Versão original deste Driver (<i>Case 1020</i>).