

Driver Altus Alnetlv2

Nome do Arquivo	Alnetlv2.dll
Fabricante	Altus Sistemas de Automação S.A.
Equipamentos	AL-2000, AL-3000, AL-600, QUARK, PICCOLO e PONTO
Protocolo	Alnet I versão 2.0
Versão	2.4.10
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	IOKit versão 1.6 ou superior
Superblocos	Sim
Nível	0

Introdução

Este é o Driver Altus Alnetlv2 para a comunicação entre sistemas da **Elipse Software** e os controladores programáveis Altus das séries AL-2000, AL-3000, AL-600, QUARK, PICCOLO e PONTO usando o protocolo Alnet I versão 2.0.

Este Driver é o resultado da unificação dos Drivers **AL2000**, **AL2000MNS**, **AL2000-Modem** e **AL2000 Timestamp**. Houve um esforço para tentar manter a compatibilidade com estas versões no que se refere às configurações dos Tags. Com exceção do Tag com o parâmetro *B2* igual a 6000, Leitura de Bloco de Dados com Estampa de Tempo, que no Driver original era identificado pelo parâmetro *B2* igual a 2000, todos os demais Tags seguem os mesmos padrões dos Drivers anteriores e podem funcionar sem alterações com aplicações já existentes para estas versões.

Embora este Driver tenha sido especificado para permitir comunicação em redes Alnet-I, também permite a comunicação em redes Alnet-II através de uma biblioteca do fabricante, que realiza a conversão entre estes dois protocolos.

NOTA

Este Driver suporta o recurso de Superblocos ou agrupamento de Tags, ou seja, a propriedade **EnableReadGrouping**.

Preparando um Equipamento

A comunicação pode se dar diretamente via cabo RS232 ou através de AL1413, utilizando um cabo RS485, para maiores distâncias. A configuração de um equipamento deve ser realizada antes usando o programa MasterTool do fabricante.

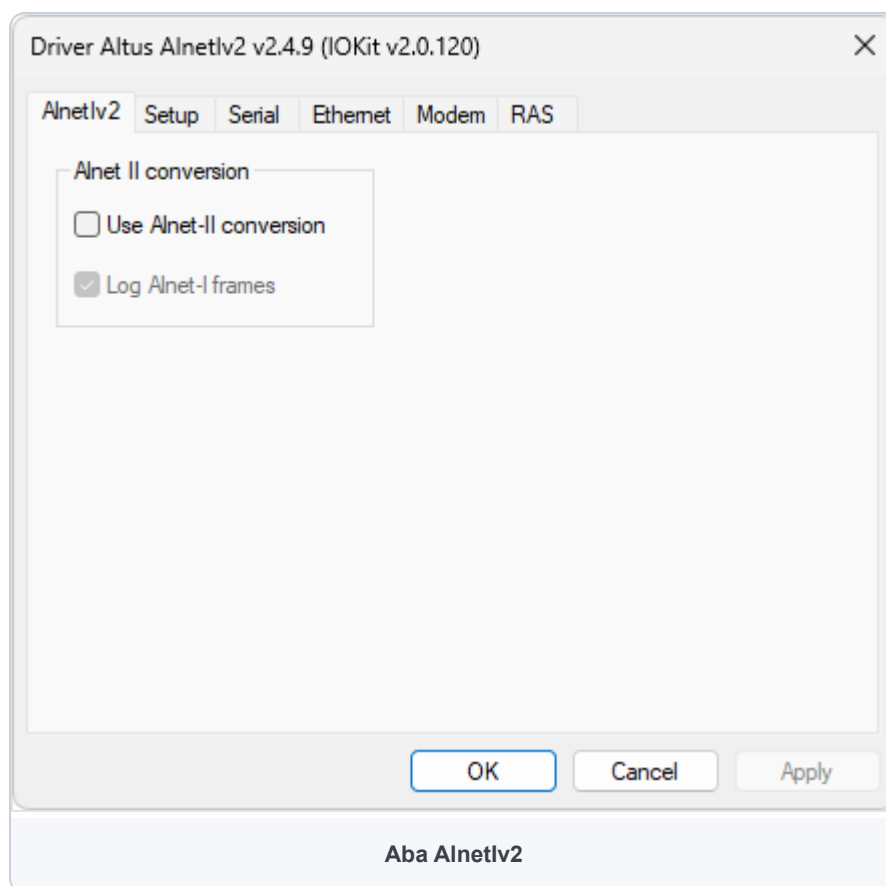
Este Driver pode também se comunicar com CLPs em rede Alnet II, se é usado um Gateway AL2400/S. Neste caso, é preciso especificar não apenas o endereço de um CLP nos Tags de Comunicação, mas também o endereço da sub-rede, multiplicado por 100 ao parâmetro *N1* ou *B1* de cada Tag. Para mais informações, consulte os tópicos Parâmetros [B] de Endereçamento para Tags Bloco e Parâmetros [N] de Endereçamento para Tags PLC. A configuração padrão de comunicação serial dos CLPs, configurável em alguns equipamentos, e deste Driver é a seguinte:

- **Velocidade de comunicação:** 9600 bps
- **Número de bits de dados por caractere:** 8 (oito)
- **Paridade:** par
- **Número de start bits:** 1 (um)
- **Número de stop bits:** 1 (um)

Configuração do Driver

Este Driver não utiliza os parâmetros [P]. Todas as configurações referentes à comunicação devem ser realizadas na janela de configurações deste Driver. Nesta janela pode-se encontrar uma aba específica, Alnetlv2, e as abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS**, específicas da biblioteca **IOKit** da **Elipse Software**, utilizada por este Driver. Para mais informações sobre esta biblioteca, consulte o tópico Documentação das Interfaces de Comunicação.

Configuração das Propriedades



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

Opções disponíveis na aba Alnetlv2

OPÇÃO	DESCRIÇÃO
Use Alnet-II conversion	Converte os pacotes de rede Alnet-I para pacotes de rede Alnet-II, permitindo a comunicação nestas redes
Log Alnet-I frames	Caso a opção Use Alnet-II conversion esteja habilitada, permite que os <i>frames</i> de pacotes de rede Alnet-I sejam visualizados nos logs antes de passarem pela camada de conversão de pacotes de rede Alnet-I para pacotes de rede Alnet-II

Referência de Tags

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

Introdução

Através dos Tags deste Driver é possível ler e escrever em operandos e tabelas de operandos de um CLP. Pode-se também acessar operandos da memória interna deste Driver, modificados por mensagens não solicitadas recebidas dos equipamentos, atribuindo-se 9999 ao parâmetro *N1* ou *B1* nos Tags que suportam mensagens não solicitadas. Este Driver, seguindo o padrão antes implementado no antigo Driver **AL2000 Timestamp**, permite acessar por mensagens não solicitadas apenas os operandos **Memória**, **Tabela de Memória**, **Octeto** e **Auxiliar**. Não é permitida a escrita de valores de mensagens não solicitadas através dos Tags. Estes valores podem apenas ser modificados por mensagens dos CLPs. Na leitura de valores de mensagens não solicitadas, a propriedade **Timestamp** de um Tag indica o momento da última modificação de um valor ou do recebimento de uma mensagem. Para mais informações sobre o recebimento de mensagens não solicitadas de um CLP, consulte o tópico Mensagens Não Solicitadas (MNS).

Os Tags de leitura e escrita de operandos podem ser configurados usando parâmetros *N* e *B* como nas versões anteriores, ou alternativamente, se este Driver é usado com o **E3**, **Power** ou **Water**, através de **Strings**. A configuração por **Strings** segue o mesmo padrão usado nos Drivers OPC, através das propriedades **Dispositivo** e **Item** dos Tags do **E3**, **Power** ou **Water**. Para mais informações, consulte o tópico Configuração por Strings.

Parâmetros [N] de Endereçamento para Tags PLC

N1	Número da rede × 100 + número de um CLP. Consulte a nota a seguir para mais informações
N2	Tipo de dados. Consulte a tabela Tipos de dados suportados para Tags PLC. Some 10000 para inteiros sem sinal, ou seja, os tipos de dados Memória e Tabela de Memória
N3	Endereço. Consulte a nota a seguir para mais informações

N4	Bit ou posição de tabela
----	--------------------------

NOTA

Alguns endereços estão expressos em octal em um CLP. Para a leitura correta destes valores nas aplicações da **Elipse Software**, os endereços devem ser convertidos para decimal.

A tabela a seguir contém uma síntese dos tipos de dados que podem ser lidos com estes Tags e a respectiva configuração, que são descritos nos tópicos a seguir.

Tipos de dados suportados para Tags PLC

TIPO DE DADOS	MNEMÔNICO	N2	N3	N4	MODO
Memória	M<end>	0 (zero)	Endereço	0: Valor, 1-8: Bit	Leitura ou escrita
Tabela de Memória	TM<end>:<pos>	1 (um)	Endereço	Posição	Leitura ou escrita
Bit de Memória	bit <n> de M<end>	2 (dois)	Endereço	<i>n</i>	Leitura ou escrita
Decimal	D<end>	3 (três)	Endereço	0: Valor, 1-8: Bit	Leitura ou escrita
Tabela de Decimais	TD<end>:<pos>	4 (quatro)	Endereço	Posição	Leitura ou escrita
Auxiliar	A<end>	5 (cinco)	Endereço	0: Valor, 1-8: Bit	Leitura ou escrita
Octeto de Entrada e Saída	E<end> e S<end>	6 (seis)	Endereço	0: Valor, 1-8: Bit	Leitura ou escrita
Inteiro de 32 Bits	I<end>	7 (sete)	Endereço	0: Valor, 1-8: Bit	Leitura ou escrita
Bit em Octeto	E<end> e S<end>	8 (oito)	Endereço	Bit (0-7)	Escrita
Bit em Auxiliar	A<end>	9 (nove)	Endereço	Bit (0-7)	Escrita

TIPO DE DADOS	MNEMÔNICO	N2	N3	N4	MODO
Real ou Float	F<end>	10	Endereço	0: Valor, 1-8: Bit	Leitura ou escrita
Tabela de Real ou Float	TF<end>:<pos>	11	Endereço	Posição	Leitura ou escrita
Tabela de Inteiro de 32 Bits	TI<end>:<pos>	12	Endereço	Posição	Leitura ou escrita
Leitura de Dados com Estampa de Tempo		13	Endereço da tabela decimal	Endereço auxiliar	Leitura
Libera Todos os Forçamentos	-	200	-	-	Escrita

Memória

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	0 (zero)
N3	Endereço
N4	Bit. Os valores possíveis são 0 : Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag lê e escreve operandos de tipo **Memória**.

Tabela de Memória

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS

N2	1 (um)
N3	Tabela
N4	Posição

Este Tag lê e escreve operandos de tipo **Tabela de Memória**.

Bit de Memória

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	2 (dois)
N3	Endereço
N4	Bit a ser acessado. O bit menos significativo é o de número 0 (zero)

Este Tag lê e escreve em bits de operandos do tipo **Bit de Memória**.

NOTA

Este Tag foi mantido por compatibilidade e deve-se evitar usá-lo em aplicações recentes. As aplicações devem utilizar o Tag de acesso a operandos de tipo Memória com o parâmetro *N4* maior que 0 (zero).

Decimal

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	3 (três)
N3	Endereço
N4	Bit. Os valores possíveis são 0 : Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag lê e escreve operandos de tipo **Decimal**.

Tabela de Decimais

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	4 (quatro)
N3	Tabela
N4	Posição

Este Tag lê e escreve em tabelas de decimais.

Auxiliar

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	5 (cinco)
N3	Endereço
N4	Bit. Os valores possíveis são 0 : Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag lê e escreve operandos de tipo **Auxiliar**.

Octeto de Entrada e Saída

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP ou 9999 para MNS
N2	6 (seis)

N3	Endereço
N4	Bit. Os valores possíveis são 0 : Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag respectivamente lê e escreve em octetos de entradas e saídas digitais.

Inteiro de 32 Bits

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	7 (sete)
N3	Endereço
N4	Bit. Os valores possíveis são 0 : Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag respectivamente lê e escreve em operandos de tipo **Inteiro de 32 bits**.

Bit em Octeto

Somente Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	8 (oito)
N3	Endereço
N4	Bit a ser acessado. O bit menos significativo é o de número 0 (zero)

Este Tag escreve em bits de octetos.

NOTA

Este Tag foi mantido por compatibilidade e deve-se evitar usá-lo em aplicações recentes. As aplicações devem utilizar o Tag de acesso a operandos de Entrada e Saída com o parâmetro *N4* maior que 0 (zero).

Bit em Auxiliar

Somente Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	9 (nove)
N3	Endereço
N4	Bit a ser acessado. O bit menos significativo é o de número 0 (zero)

Este Tag respectivamente lê e escreve em bits de operandos de tipo **Auxiliar**.

NOTA

Este Tag foi mantido por compatibilidade e deve-se evitar usá-lo em aplicações recentes. As aplicações devem utilizar o Tag de acesso a operandos de tipo Auxiliar com o parâmetro *N4* maior que 0 (zero).

Real ou Float

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	10
N3	Endereço
N4	Bit. Os valores possíveis são 0 : Lê o valor de um operando inteiro ou maior que 0 (zero) para ler apenas o bit especificado. O bit menos significativo é o de número 1 (um)

Este Tag respectivamente lê e escreve em operandos de tipo **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Tabela de Real ou Float

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	11
N3	Tabela
N4	Posição

Este Tag lê e escreve em tabelas de **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Tabela de Inteiro de 32 Bits

Leitura e Escrita

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	12
N3	Tabela
N4	Posição

Este Tag lê e escreve em tabelas de **Inteiros de 32 bits**.

Leitura de Dados com Estampa de Tempo

Leitura

N1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
N2	13

N3	Endereço de uma tabela de leitura decimal
N4	Endereço auxiliar

Este Tag lê variáveis de tabelas de decimais com estampa de tempo, com intervalo constante entre variáveis. Para mais informações sobre a utilização deste recurso, consulte o tópico [Leitura de Dados com Estampa de Tempo](#).

Parâmetros [B] de Endereçamento para Tags Bloco

A tabela a seguir contém os tipos de dados que podem ser lidos com estes Tags e a respectiva configuração, que são descritos nos tópicos a seguir.

Tipos de dados suportados para Tags Bloco

TIPO DE DADOS	B2	B3	B4	MODOS
Tabela de Memória	Número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Tabela de Decimal	1000 + número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Status de Barramentos de Entrada e Saída	2000	Número do barramento	-	Leitura
Status de Comunicação	2001	Reset de variáveis	-	Leitura
Status do Equipamento	2002	-	-	Leitura
Bloco de Memória	3000	Endereço inicial	-	Leitura ou escrita
Tabela de Real ou Float	4000 + número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Tabela de Inteiro de 32 Bits	6000 + número da tabela	Posição inicial da tabela	-	Leitura ou escrita
Bloco de Real ou Float	5000	Endereço inicial	-	Leitura ou escrita

TIPO DE DADOS	B2	B3	B4	MODO
Bloco de Inteiro de 32 Bits	5003	Endereço inicial	-	Leitura ou escrita
Bloco de Dados com Estampa de Tempo	5005	Posição inicial da tabela decimal	Endereço auxiliar	Leitura

Tabela de Memória

Leitura e Escrita

B1	Endereço de um CLP. Os valores possíveis são Número da rede $\times$ 100 + Número de um CLP ou 9999 para MNS
B2	Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de memória.

Tabela de Decimal

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede $\times$ 100 + Número de um CLP
B2	1000 + Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de decimal.

Status de Barramentos de Entrada e Saída

Somente Leitura

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	2000
B3	Número do barramento
B4	Não usado

Este Tag lê o status de um barramento de entrada e saída e dos módulos que o compõem. Os significados dos valores dos Elementos de Bloco retornados estão descritos na tabela a seguir.

Elementos de Bloco de um barramento de entrada e saída

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
1	TOT_BAR	Total de barramentos de entrada e saída possíveis em um CLP
2	TIPO_BAR	Tipo de um barramento. Os valores possíveis são 0 : Barramento AL-1000, 1 : Barramento AL-3000 ou 2 : Barramento PONTO
3	ST_BAR	Estado de um barramento
4	NUM_MOD	Número de módulos de entrada e saída possíveis no barramento solicitado
5	TIPO_MOD_0	Tipo do módulo 0 (zero)
6	ST_MOD_0	Estado do módulo 0 (zero)
7	TIPO_MOD_1	Tipo do módulo 1 (um)
8	ST_MOD_1	Estado do módulo 1 (um)
...	...	...
	TIPO_MOD_N	Tipo do módulo <i>n</i>
	ST_MOD_N	Estado do módulo <i>n</i>

Status de Comunicação

Somente Leitura

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	2001
B3	<i>Reset</i> de variáveis. Os valores possíveis são 0 : Os valores dos dados sobre recepção e transmissão não são alterados ou 1 : Os valores dos dados sobre recepção e transmissão são zerados
B4	Não usado

Este Tag lê informações sobre o estado da comunicação ALNET II. Os significados dos valores dos Elementos de Bloco estão descritos na tabela a seguir.

Elementos de Bloco do status da comunicação

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
1	DADOS_TX	Número de transmissões sem erros
2		Número de transmissões com erro de colisão
3		Número de transmissões com erro de <i>underrun</i>
4		Número de transmissões sem recepção de ACK de hardware
5		Número de transmissões canceladas por esgotamento de retentativas
6		Número de <i>time-outs</i> de serviço
7		Número de faltas de <i>buffers</i> de transmissão
8		Reservado
9		Reservado
10	DADOS_RX	Número de recepções sem erros
11		Número de recepções com erro de colisão

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
12		Número de recepções com erro de <i>overrun</i>
13		Número de recepções com erro de CRC
14		Número de recepções com erro de alinhamento
15		Número de recepções de pacotes com erro de tamanho
16		Número de <i>time-outs</i> de pacote
17		Número de faltas de <i>buffers</i> de recepção
18		Reservado
19	DADOS_COM	Velocidade de comunicação
20		Endereço de um nó
21		Endereço da sub-rede local
22		Grupos de <i>multicast</i> dos quais um nó é parte
23		Reservado
24		Reservado
25		<i>Time-out</i> intrabarramento
26		<i>Time-out</i> interbarramento
27		<i>Time-out</i> de pacote
28		Número máximo de retentativas de transmissão
29		Identificador de um nó na rede, com 20 bytes
30	DADOS_CNX	Tipo de conexão física. Os valores possíveis são 0 : Elétrica padrão ou 1 : Óptica

ELEMENTO	DESCRIÇÃO	COMENTÁRIO
31		Redundância de conexões físicas. Os valores possíveis são 0 : Sem redundância ou 1 : Com redundância
32		Período para teste da redundância, em segundos
33		Tempo para a comutação da conexão física, em segundos
34		Reservado
35		Reservado
36		Conexão física selecionada. Os valores possíveis são 1 (um) ou 2 (dois)
37		Estado da conexão 1 (um). Os valores possíveis são 0 : OK ou 1 : Falha
38		Estado da conexão 2 (dois). Os valores possíveis são 0 : OK ou 1 : Falha
39		Reservado
40		Conexão física forçada. Os valores possíveis são 0 (sem forçamento), 1 (um) ou 2 (dois)
41		Estado forçado da conexão 1 (um). Os valores possíveis são 0 : OK ou 1 : Falha
42		Estado forçado da conexão 2 (dois). Os valores possíveis são 0 : OK ou 1 : Falha

Status de um Equipamento

Somente Leitura

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	2002
B3	Não usado
B4	Não usado

Este Tag lê informações de estado de um equipamento. Os significados dos valores de cada Elemento de Bloco estão descritos na tabela a seguir.

Elementos de Bloco do status de um equipamento

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
1	MODELO_UCP	Modelo da UCP
2	VERS_EXE_H	Versão do executável
3	VERS_EXE_L	
4	MODO_OP_1	Modo de operação 1 (um)
5	COD_MESG_1	Código de mensagem 1 (um)
6	RAM_LIV_2H	Espaço livre da RAM da aplicação banco 2 (dois)
7	RAM_LIV_2H	
8	RAM_LIV_1H	Banco 1 (um)
9	RAM_LIV_1L	
10	STAT_RAM_1	Estado da RAM da aplicação 1 (um)
11	TC_INST_H	Tempo de ciclo instantâneo
12	TC_INST_L	
13	TC_MED_H	Tempo de ciclo médio
14	TC_MED_L	
15	TC_MAX_H	Tempo de ciclo máximo
16	TC_MAX_L	
17	TC_MIN_H	Tempo de ciclo mínimo
18	TC_MIN_L	

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
19	PER_E018	Período de chamada do módulo E018
20	PER_E019	Período de chamada do módulo E019
21	-	-
22	TEMPO_PROG	Tempo máximo de execução de uma aplicação
23	STAT_RAM_2	Estado da RAM da aplicação 2 (dois)
24	MODO_OP_2	Modo de operação 2 (dois)
25	RAM_LIV_8H	Espaço livre da RAM da aplicação banco 8 (oito)
26	RAM_LIV_8L	
27	RAM_LIV_7H	Banco 7 (sete)
28	RAM_LIV_7L	
29	RAM_LIV_6H	Banco 6 (seis)
30	DA RAM_LIV_6L	
31	RAM_LIV_5H	Banco 5 (cinco)
32	RAM_LIV_5L	
33	RAM_LIV_4H	Banco 4 (quatro)
34	RAM_LIV_4L	
35	RAM_LIV_3H	Banco 3 (três)
36	RAM_LIV_3L	
37	STATUS_EPR	Estado da EPROM de uma aplicação
38	EPR_LIV_8H	Espaço livre da EPROM da aplicação banco 8 (oito)

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
39	EPR_LIV_8L	
40	EPR_LIV_7H	Banco 7 (sete)
41	EPR_LIV_7L	
42	EPR_LIV_6H	Banco 6 (seis)
43	EPR_LIV_6L	
44	EPR_LIV_5H	Banco 5 (cinco)
45	EPR_LIV_5L	
46	EPR_LIV_4H	Banco 4 (quatro)
47	EPR_LIV_4L	
48	EPR_LIV_3H	Banco 3 (três)
49	EPR_LIV_3L	
50	EPR_LIV_2H	Banco 2 (dois)
51	EPR_LIV_2L	
52	EPR_LIV_1H	Banco 1 (um)
53	EPR_LIV_1L	
54	COD_MESG_2	Código de mensagem 2 (dois)
55	COD_MESG_3	Código de mensagem 3 (três)
56	COD_MESG_4	Código de mensagem 4 (quatro)
57	IDENT_0	Identificação secundária de produto (ASCII)
58	IDENT_1	

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
59	IDENT_2	
60	IDENT_3	
61	IDENT_4	
62	IDENT_5	
63	IDENT_6	
64	IDENT_7	
65	EPR_LIV_16H	Banco 16
66	EPR_LIV_16L	
67	EPR_LIV_15H	Banco 15
68	EPR_LIV_15L	
69	EPR_LIV_14H	Banco 14
70	EPR_LIV_14L	
71	EPR_LIV_13H	Banco 13
72	EPR_LIV_13L	
73	EPR_LIV_12H	Banco 12
74	EPR_LIV_12L	
75	EPR_LIV_11H	Banco 11
76	EPR_LIV_11L	
77	EPR_LIV_10H	Banco 10
78	EPR_LIV_10L	

ELEMENTO	MNEMÔNICO	DESCRIÇÃO
79	EPR_LIV_9H	Banco 9 (nove)
80	EPR_LIV_9L	

Bloco de Memória

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	3000
B3	Endereço inicial
B4	Não usado

Este Tag lê e escreve em blocos de operandos do tipo **Memória**.

Tabela de Real ou Float

Leitura e Escrita

B1	Endereço de um CLP
B2	4000 + Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de operandos de tipo **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Tabela de Inteiro de 32 Bits

Leitura e Escrita

B1	Endereço de um CLP
B2	6000 + Número da tabela
B3	Posição
B4	Não usado

Este Tag lê e escreve em tabelas de operandos de tipo **Inteiro de 32 bits**.

Bloco de Real ou Float

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede $\times$ 100 + Número de um CLP
B2	5000
B3	Endereço inicial
B4	Não usado

Este Tag lê e escreve em blocos de operandos do tipo **Real** ou **Float**.

NOTA

Para valores no formato **Float** com números não normalizados, assume-se o valor 0 (zero) para os casos de *underflow* e assume-se o valor **null** para os casos de *overflow*.

Bloco de Inteiro de 32 Bits

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede $\times$ 100 + Número de um CLP
B2	5003

B3	Endereço inicial
B4	Não usado

Este Tag lê e escreve em blocos de operandos do tipo **Inteiro de 32 bits**.

Bloco de Dados com Estampa de Tempo

Leitura e Escrita

B1	Endereço de um CLP, no formato Número da rede × 100 + Número de um CLP
B2	5005
B3	Posição inicial da tabela decimal
B4	Endereço auxiliar

Este Tag permite a realização da leitura de variáveis de tabelas decimais com estampa de tempo. Para mais informações sobre a utilização deste recurso, consulte o tópico [Leitura de Dados com Estampa de Tempo](#).

Configuração por Strings

Este Driver permite a configuração de Tags através de **Strings**, utilizando o mesmo formato usado no Driver OPC. A configuração por **Strings** é realizada através das propriedades **Dispositivo** e **Item** dos Tags do **E3**, **Power** ou **Water**.

NOTA

A configuração por **Strings** ainda não estava disponível para o **Elipse SCADA** até a publicação desta versão.

A propriedade **Dispositivo** pode ser utilizada para definir o endereço de um CLP, usando o formato **<net>.plc:**, onde **net** é um parâmetro opcional identificando o número da rede, e **plc** é o endereço de um CLP. Se a propriedade **Dispositivo** é deixada em branco, o endereço deve ser informado na propriedade **Item** utilizando o mesmo formato, seguido por dois pontos. Se o endereço de rede não está presente, considera-se como 0 (zero). Se o endereço é definido na propriedade **Dispositivo**, os dois pontos finais podem ser dispensados. A propriedade **Item** define os dados do operando a ser acessado. A tabela a seguir contém os mnemônicos que podem ser utilizados com este Driver.

Mnemônicos suportados

MNEMÔNICO	DESCRIÇÃO	MODO
<<net>.plc:>Ax	Auxiliar	Leitura e escrita

MNEMÔNICO	DESCRIÇÃO	MODO
<<net>.plc:>Ax.b	Bit de auxiliar	Leitura e escrita
<<net>.plc:>Dx	Decimal	Leitura e escrita
<<net>.plc:>Dx.b	Bit de decimal	Leitura e escrita
<<net>.plc:>Ex	Entrada	Leitura
<<net>.plc:>Ex.b	Bit de entrada	Leitura
<<net>.plc:>Fx	Real (<i>Float</i>)	Leitura e escrita
<<net>.plc:>Ix	Inteiro (<i>32-bits</i>)	Leitura e escrita
<<net>.plc:>Mx	Memória	Leitura e escrita
<<net>.plc:>Mx.b	Bit de memória	Leitura e escrita
<<net>.plc:>Sx	Saída	Escrita
<<net>.plc:>Sx.b	Bit de saída	Escrita
<<net>.plc:>TDx.y	Tabela de decimais	Leitura e escrita
<<net>.plc:>TFx.y	Tabela de Reais (<i>Float</i>)	Leitura e escrita
<<net>.plc:>TIx.y	Tabela de Inteiros (<i>32-bits</i>)	Leitura e escrita
<<net>.plc:>TMx.y	Tabela de memórias	Leitura e escrita
<<net>.plc:>BSx	Status do barramento de entrada e saída	Leitura
<<net>.plc:>CSx	Status da comunicação	Leitura
<<net>.plc:>ES	Status do equipamento	Leitura

Em que:

- Os caracteres em **negrito** representam mnemônicos fixos que identificam os comandos, enquanto os caracteres que não estão em **negrito** representam os valores que devem ser substituídos

- O caractere *y* representa a posição inicial na tabela
- O caractere *b* representa o número de um bit, variando de 0 (zero, bit menos significativo) até 31 (bit mais significativo), porém o número máximo depende do número de bits presentes no tipo de dados

A seguir alguns exemplos de uso dos mnemônicos e a respectiva conversão em parâmetros *N* ou *B*, conforme realizada internamente por este Driver.

Exemplos de uso de mnemônicos com Tags do tipo PLC

DISPOSITIVO	ITEM	N1 OU B1	N2 OU B2	N3 OU B3	N4 OU B4
	1:D8	1 (um)	3 (três)	8 (oito)	0 (zero)
	2.5:F50	205	10	50	0 (zero)
8.2:	NaN	802	6 (seis)	7 (sete)	0 (zero)
	120:S5	120	6 (seis)	5 (cinco)	0 (zero)
5.4:	A120	504	5 (cinco)	120	0 (zero)
	2:M8	2 (dois)	0 (zero)	8 (oito)	0 (zero)
	6.5:TD6.123	605	4 (quatro)	6 (seis)	123
2.3	TF84.5	203	11	84	5 (cinco)
6	TM5.64	6 (seis)	1 (um)	5 (cinco)	64
7.1:	M125.1	701	0 (zero)	125	2 (dois)

NOTA

Os mnemônicos **E** e **S** são ambos mapeados para Tags PLC com o parâmetro *N2* igual a 6 (seis). A diferença está apenas na operação em si, pois somente são permitidas leituras em operandos **E** e escritas em operandos **S**.

Leitura de Dados com Estampa de Tempo

Este Driver permite a leitura de dados com estampa de tempo vinda de um CLP indicando o momento da aquisição.

Os modos de aquisição são a leitura de blocos de dados adquiridos simultaneamente, ou seja, com a mesma estampa de tempo, e a leitura de blocos de dados adquiridos em intervalos constantes. Em ambos os casos, o programa dos controladores precisa ser capaz de atender a determinados requisitos. A seguir são detalhados estes modos de trabalho.

Modo de Blocos de Dados com Aquisição Simultânea

Neste modo, pode-se realizar a leitura de variáveis de tabelas decimais com estampa de tempo quando se tem um conjunto de variáveis armazenadas simultaneamente. Logo, cada conjunto de valores com a informação de tempo associada formam uma estrutura, que é repetida diversas vezes, formando assim uma tabela. Cada estrutura deve estar no formato descrito a seguir.

```
Dado 1 - HHMMSS (Exemplo: 19:02 e 34 segundos são colocados como 190234)
Dado 2 - DDMMAA (Exemplo: 18 de fevereiro de 1998 é colocado como 180298)
Dado 3 - valor 1
Dado 4 - valor 2
(...)
Dado N - valor N - 2
```

O tamanho máximo permitido para uma tabela é de 255 operandos inteiros, ou 1020 bytes ao todo.

Por exemplo, em uma tabela de 255 posições, deseja-se armazenar várias leituras de 5 (cinco) variáveis analógicas com a estampa de tempo. Logo, a tabela após preenchida tem a configuração da tabela a seguir, supondo que a data atual é 18/02/1998, 9:00:00, com aquisições de minuto em minuto.

Exemplo de estrutura de dados com estampa de tempo

POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR
0	90000	7	90100	14	90200
1	180298	8	180298	15	180298
2	Dado	9	Dado	16	Dado
3	Dado	10	Dado	17	Dado
4	Dado	11	Dado	18	Dado
5	Dado	12	Dado	19	Dado
6	Dado	13	Dado	20	Dado

E assim por diante, até preencher a tabela.

NOTA

Como tem-se apenas 2 (dois) dígitos para o ano, este Driver está preparado para funcionamento até o ano de 2098.

Para o controle de acesso à tabela, existem 4 (quatro) bits de uma memória auxiliar em um CLP:

- **BIT 0 (pede_permissao_de_leitura):** Executa uma requisição de acesso à tabela para um CLP

- **BIT 1 (leitura_permitida):** Verifica se a requisição foi aceita
- **BIT 2 (limpa):** Este Driver escreve o valor 1 (um) neste bit e o CLP se encarrega de zerar a tabela e voltar este bit para 0 (zero)
- **BIT 3 (tabela_zerada):** Este bit informa se a tabela está zerada

Para a leitura dos dados armazenados, deve-se utilizar um Tag Bloco com o parâmetro *B2* igual a 6000, em que o parâmetro *B3* indica a posição inicial da tabela e o parâmetro *B4* indica o endereço do byte contendo os bits auxiliares descritos anteriormente.

Na leitura, caso exista mais de um conjunto de valores ou estruturas na tabela, este Driver devolve uma lista de valores para uma aplicação da **Elipse Software**, como é o caso do exemplo anterior. A leitura de estruturas prossegue até que este Driver encontre valores nulos nos primeiros 2 (dois) campos de data e hora, indicando o fim da tabela, ou até que ultrapasse o limite de 255 operandos.

A utilização dos recursos do exemplo anterior está prevista somente para a utilização com blocos. Portanto, no caso de apenas uma variável, crie um bloco de apenas uma variável.

Sequência de Passos Executados por este Driver

1. Verifica o bit 3 (três, tabela zerada). Se está configurado em 1 (um), sai ou prossegue em caso contrário.
2. Escreve 1 (um) no bit 0 (zero, pede permissão).
3. Verifica o bit 1 (um). Se está configurado em 1 (um), prossegue ou verifica mais uma vez em caso contrário. Persistindo a não autorização, sai. Neste caso, o CLP deve tratar a requisição pendente, voltando o bit 0 (zero, pede permissão) para 0 (zero).
4. Lê as primeiras $n + 2$ (duas) variáveis, em que n é o tamanho do bloco.
5. Caso os 2 (dois) primeiros valores contenham uma data válida, estes valores são copiados.
6. Repete o passo anterior até chegar ao fim da tabela ou quando existirem valores nulos nos 2 (dois) primeiros campos.
7. Zera a tabela, caso tenha sido lida pelo menos 1 (uma) estrutura, escrevendo 1 (um) no bit 2 (dois, limpa).
8. Este Driver devolve os valores para uma aplicação da **Elipse Software**.

Modo de Blocos de Dados com Aquisição a Intervalos Constantes

Deve-se utilizar este modo quando o intervalo de tempo entre os dados em uma tabela é fixo. Assim pode-se economizar espaço em memória dentro de um CLP, armazenando-se somente o instante inicial da coleta. Este modo só pode ser usado quando há vários dados para 1 (uma) só variável. Para isto, deve-se realizar o procedimento a seguir.

Em um CLP, deve existir um programa que armazene os dados em uma tabela de decimais. O formato da tabela tem a estrutura a seguir.

```
Dado 1 - HHMMSS
Dado 2 - DDMMAA
Dado 3 - intervalo de tempo entre cada coleta, em milissegundos
Dado 4 - valor 1
(...)
Dado N - valor N - 3
```

Por exemplo, em uma tabela de 255 posições, deseja-se armazenar várias leituras de 1 (uma) variável analógica com intervalos de tempo constantes. Logo, a tabela após preenchida tem com a seguinte configuração, supondo que a data atual é 18/02/1998, 9:00:00, com aquisições de segundo em segundo.

Exemplo de dados com intervalo de estampa de tempo constante

POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR	POSIÇÃO NA TABELA	VALOR
0	90000	7	Dado	14	Dado
1	180298	8	Dado	15	Dado
2	1000	9	Dado	16	Dado
3	dado	10	Dado	17	Dado
4	dado	11	Dado	18	Dado
5	dado	12	Dado	19	Dado
6	dado	13	Dado	20	Dado

E assim por diante, até preencher a tabela. Para o controle de acesso à tabela, existem 4 (quatro) bits de uma memória auxiliar em um CLP. Este auxiliar é informado na leitura de um Tag.

- **BIT 0 (pede_permissao_de_leitura):** Executa uma requisição de acesso à tabela para um CLP
- **BIT 1 (leitura_permitida):** Verifica se a requisição foi aceita
- **BIT 2 (limpa):** Este Driver escreve o valor 1 (um) neste bit e o CLP se encarrega de zerar a tabela e voltar este bit para 0 (zero)
- **BIT 3 (tabela_zerada):** Este bit informa se a tabela está zerada

A leitura dos dados só pode ser realizada através de um Tag PLC com o parâmetro *N2* igual a 13, em que o parâmetro *N3* indica o endereço da tabela decimal e o parâmetro *N4* indica o endereço do auxiliar que contém os bits de controle. Consulte as tabelas Tipos de dados suportados para Tags PLC e Tipos de dados suportados para Tags Bloco para mais informações. Caso o valor de um dado lido seja diferente de 0 (zero), este Driver devolve uma lista de valores de eventos para uma aplicação da **Elipse Software**, cada evento com uma estampa de tempo igual à estampa de tempo da tabela + $(N - 3) \times$ intervalo de tempo, em que *N* é a posição da variável dentro da tabela.

Sequência de Passos Executados por este Driver

Ao executar a leitura da tabela com a configuração anterior, este Driver executa os passos a seguir.

1. Verifica o bit 3 (três, tabela zerada). Se está configurado em 1 (um) sai ou prossegue em caso contrário.
2. Escreve 1 (um) no bit 0 (zero, pede permissão).

3. Verifica o bit 1 (um) e, se está configurado em 1 (um), prossegue. Caso contrário verifica mais uma vez. Persistindo a não autorização, sai. Neste caso o CLP deve tratar a requisição pendente, voltando o bit 0 (zero, pede permissão) para 0 (zero).
4. Lê as primeiras 3 (três) variáveis com a estampa de tempo e o intervalo.
5. Caso os 3 (três) primeiros valores contenham uma data válida, esta é armazenada.
6. Lê as próximas posições da tabela, de 12 em 12 unidades, armazenando o valor juntamente com a estampa de tempo adicionada de cada intervalo sucessivamente. Caso o primeiro valor lido de cada leitura seja igual a 0 (zero), devolve os últimos valores lidos e zera a tabela, ou seja, só transfere valores de um bloco de 12 variáveis se o primeiro valor é diferente de 0 (zero).
7. Zera a tabela, caso tenha sido lida pelo menos uma estrutura, e escreve 1 (um) no bit 2 (dois, limpa).
8. Este Driver devolve os valores para uma aplicação da **Elipse Software**.

Mensagens Não Solicitadas (MNS)

O uso das mensagens não solicitadas permite melhorar significativamente a performance de uma aplicação, pois não há necessidade de executar *polling* nas variáveis de um PLC o tempo todo, mesmo quando estas variáveis não sofrem alteração. Ao contrário, usando-se mensagens não solicitadas, toda vez que uma variável sofre alteração, o próprio PLC envia uma mensagem avisando este Driver sobre o novo valor que, por sua vez, altera a base de dados.

Uma aplicação, por sua vez, deve configurar os Tags PLC e Bloco de maneira muito semelhante àquela usada para Tags normais, por *polling*, mudando-se apenas o parâmetro *N1* ou *B1* para 9999. Este número indica a este Driver que os valores destes Tags devem ser lidos não de um PLC, mas diretamente da base de dados deste Driver, atualizada pelas MNS. Os Tags de uma aplicação, portanto, continuam funcionando por *polling*, exceto que a leitura é direta de uma memória imagem contida neste Driver, que não gera comunicação com um PLC. Como esta operação é extremamente rápida, sugere-se usar um período de *scan* muito curto, da ordem de milissegundos, o que permite uma atualização imediata dos Tags em caso de alteração do valor por mensagens não solicitadas. A leitura de mensagens não solicitadas só está habilitada neste Driver para os operandos do tipo **Memória**, **Tabela de Memória**, **Auxiliar** e **Octeto**.

Exemplo de Tag PLC para Mensagens Não Solicitadas

- **N1:** 101 (Rede igual a um e PLC igual a um)
- **N2:** 0 (zero, Memória)
- **N3:** 1 (um, Memória M1)
- **N4:** 0 (zero, não usado neste caso)
- **Scan:** 500 (500 milissegundos)

Para passar este Tag PLC para leitura diretamente da memória imagem deste Driver, altere o parâmetro *N1* para 9999. Uma vez que a operação neste caso é bem mais rápida do que por *polling*, pode-se alterar também o período de *scan* para apenas alguns milissegundos. A nova configuração é a seguinte:

- **N1:** 9999 (Leitura da Memória Imagem)
- **N2:** 0 (zero, Memória)
- **N3:** 1 (um, Memória M1)
- **N4:** 0 (zero, não usado neste caso)

- **Scan:** 10 (10 milissegundos)

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **Alnetlv2**.

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 das Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.4.10	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (Case 37935).
2.4.9	02/10/2020	C. Mello	• Adicionado suporte para operandos do tipo Inteiro de 32 Bits (Case 29471). • Alterado o parâmetro B2 do Tag Bloco de Dados com Estampa de Tempo de 6000 para 5005.
2.4.8	27/09/2019	C. Mello	• Driver portado para o Visual Studio 2017 (Case 27437).
2.4.7	27/09/2013	A. Quites	• Ajustes na validação de parâmetros dos Tags (Case 14706). • Corrigido um erro na leitura de Tags internos da biblioteca IOKit (Case 13585). • Driver portado para a biblioteca IOKit versão 2 (Case 13654).
2.3.1	07/10/2008	A. Quites	• Adicionado o tipo de dados inteiro sem sinal (Case 9555).
2.2.1	31/10/2007	A. Quites	• Corrigido um erro na escrita de bits de operandos Auxiliar e Octeto com o parâmetro N4 maior que 0 (zero) (Case 8887).
2.1.1	21/03/2007	A. Quites	• Corrigido um erro na leitura de blocos de dados com estampa de tempo, adquiridos no modo de aquisição simultânea, em que este Driver não

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			enviava um comando para limpar a tabela após a leitura, de forma que os dados eram lidos mais de uma vez (<i>Case 7374</i>).
2.0.1	28/07/2006	A. Quites	• Versão original para a biblioteca IOKit , unificando os Drivers AL2000 , AL2000 Timestamp , AL2000MNS e AL2000-Modem .
1.0.1		R. Haetinger	• Todas as publicações de Drivers para Altus AL2000 anteriores ao controle de revisões.