

Driver Power Measurements ION

Nome do Arquivo	ION32.dll
Fabricante	Power Measurements
Equipamentos	
Protocolo	ION versão 2.0
Versão	2.0.6
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	Nenhuma
Leitura com Superblocos	Não
Nível	31202

Introdução

Este Driver comunica-se com os modelos 7500, 7600, 8500 e 8400 dos medidores ION da Power Measurements, bem como outros medidores com suporte ao protocolo ION versão 2.0. Os dados extraídos dos medidores da Power Measurements incluem dados em tempo real, eventos, *data recorders* e *waveforms* (formatos de onda).

Configuração do Driver

Esta seção contém informação sobre os parâmetros de configuração deste Driver.

Parâmetros [P]

Este Driver não utiliza os parâmetros [P] de configuração. Todas as configurações deste Driver e da biblioteca **IOKit** devem ser realizadas na Janela de Configuração.

Janela de Configuração do Driver

Driver Power Measurements ION v2.0.5 (IOKit v2.0.110)

ION Setup Serial Ethernet Modem RAS

Master Address: 200

Default Slave Address: 100

☐ Remove Duplicated FF bytes

☐ Data Recorder Exceptions return NULL

☒ Use User Security Level

User Number (1 to 16): 1 Password:

OK Cancel Apply

Aba ION

As opções disponíveis na aba **ION** estão descritas na tabela a seguir.

Opções disponíveis na aba ION

OPÇÃO	DESCRIÇÃO
Master Address	Endereço de origem, ou seja, o endereço que este Driver assume na rede. Não deve conflitar com os endereços dos medidores presentes na rede (<i>Slave Id</i>)
Default Slave Address	Identifica o endereço de um medidor se o parâmetro <i>N1</i> ou <i>B1</i> de um Tag está configurado com o valor 0 (zero)
Remove Duplicated FF bytes	Remove os bytes FF indesejados que aparecem no início de um <i>frame</i> como resultado de chaveamento da portadora se um canal de comunicação dedicado apresenta ruído
Data Recorder Exceptions return NULL	Retorna NULL e não o texto ou descrição de uma exceção informado por um medidor durante a extração de memória de um registro de dados, ou leitura de

OPÇÃO	DESCRIÇÃO
	memória de massa
Use User Security Level	Configura este Driver para o controle de senha alfanumérica dos usuários
User Number (1 to 16)	Número do usuário, entre 1 (um) e 16, utilizado por este Driver se a opção Use User Security Level está habilitada
Password	Senha alfanumérica, com até 6 (seis) caracteres, para o usuário informado na opção User Number (1 to 16)

Para mais informações sobre a configuração das abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS**, consulte o tópico Documentação das Interfaces de Comunicação.

Parâmetros Offline

Todos os parâmetros descritos no tópico Janela de Configuração do Driver podem ser modificados em tempo de execução usando os Tags da biblioteca **IOKit**. Para isto, utilize as **Strings** a seguir:

- ION.MasterAddress (opção Master Address)
- ION.DefaultSlaveAddress (opção Default Slave Address)
- ION.CutDoubleFF (opção Remove Duplicate FF bytes)
- ION.NullDRExcept (opção Data Recorder Exceptions return NULL)
- ION.UserPwd (opção Password)
- ION.UserNumber (opção User Number)
- ION.UserSecurityLevel (opção Use User Security Level)

Referência de Tags

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

Parâmetros [N]

Esta seção contém informações sobre a configuração dos parâmetros **[N]** deste Driver. Todos os Tags utilizam o parâmetro *N1* para identificar o endereço de um medidor ou o parâmetro *N1* igual a 0 (zero) se é utilizado o endereço padrão definido na Janela de Configuração deste Driver.

Sync

Somente Escrita

N1	Endereço de um medidor (<i>Slave Id</i>)
N2	0 (zero)
N3	0 (zero)
N4	0 (zero)

Este Tag escreve o valor do deslocamento GMT (*Greenwich Mean Time*), em segundos. Este valor é adicionado ao valor do relógio interno de um computador e então enviado a um medidor para sincronização.

Index and Range Definitions for Log Table

Leitura ou Escrita

N1	Endereço de um medidor (<i>Slave Id</i>)
N2	4 (quatro)
N3	<i>Config</i>
N4	Faixa de variação

Durante a leitura de logs, uma estrutura de variação (*Range*) é reservada para cada tabela, com no máximo 20 tabelas simultâneas. Cada estrutura possui algumas variáveis que auxiliam na extração de dados de cada tabela. Estes valores variam entre 0 (zero) e 19, um para cada tabela ou *waveform*, conforme a tabela a seguir.

Valores possíveis para Config

VALOR	DESCRIÇÃO
0	Registro inicial de leitura (leitura ou escrita)
2	<i>Handle</i> do ponteiro de uma tabela (leitura ou escrita)
3	Método do ponteiro de uma tabela, geralmente igual a 28h (leitura ou escrita)

VALOR	DESCRIÇÃO
4	Posição do ponteiro de uma tabela, atualizado com um medidor
5	Quantidade de registros retornados em cada leitura, até o máximo de 15 registros (leitura ou escrita)
6	Atualização de uma tabela. Os valores possíveis são 1 (um) quando o registro atinge o ponteiro de uma tabela ou 0 (zero) caso contrário

Password

Somente Escrita

N1	Endereço de um medidor (<i>Slave Id</i>)
N2	5 (cinco)
N3	0 (zero)
N4	0 (zero)

Número inteiro de 32 bits sem sinal utilizado para escrever qualquer valor em um medidor.

Read Waveforms

Somente Leitura

N1	Endereço de um medidor (<i>Slave Id</i>)
N2	9 (nove)
N3	<i>Handle</i>
N4	Faixa de variação

Este Tag retorna todos os valores de um log de formato de onda (*waveform*). Cada log armazena formatos de onda para uma variável.

NOTA

Este comando leva um tempo considerável para ser completado.

Waveforms Read Status

Somente Leitura

N1	Endereço de um medidor (<i>Slave Id</i>)
N2	10
N3	<i>Handle</i>
N4	Faixa de variação

Este Tag retorna o valor -1 (menos um) se a operação com o Tag Read Waveforms apresenta uma falha, o valor 0 (zero) se o formato de onda (*waveform*) não contém nenhum valor ou o valor 1 (um) se há algum formato de onda disponível.

Get Waveforms

Somente Leitura

N1	Endereço de um medidor (<i>Slave Id</i>)
N2	11
N3	<i>Handle</i>
N4	Faixa de variação

Este Tag retorna a informação para uma aplicação quando um formato de onda (*waveform*) está disponível, conforme o Tag Waveforms Read Status que retorna o valor 1 (um). Um formato de onda é uma sequência de números para um mesmo Tag com a propriedade **TimeStamp** alterada para cada valor.

Parâmetros [B]

Esta seção contém informações sobre a configuração dos parâmetros **[B]** deste Driver. Todos os Tags utilizam o parâmetro *B1* para identificar o endereço de um medidor ou o parâmetro *B1* igual a 0 (zero) se é utilizado o endereço padrão definido na Janela de Configuração deste Driver.

Simple Method

Leitura ou Escrita

B1	Endereço de um medidor (<i>Slave Id</i>)
B2	1 (um)
B3	<i>Handle</i>
B4	Método

Operação de leitura ou escrita utilizando o Método Simples. Para o endereço do *handle* (parâmetro *B3*), consulte o documento **Module and Register Handles for ION Meters**, disponível com este Driver. Para o método (parâmetro *B4*), consulte a tabela a seguir.

Handles do Método Simples

CÓDIGO	NOME
01h	Read ION class
02h	Read ION name
03h	Read ION label
14h	Read register time
15h	Read register value
1Ch	Read Boolean register on name
1Dh	Read Boolean register off name
1Eh	Read Boolean register on label
1Fh	Read Boolean register off label
20h	Read Boolean register state label
23h	Read Array register depth

CÓDIGO	NOME
25h	Read Array register rollover
28h	Read log register position
2Dh	Read event log alarms
50h	Read module setup counter
64h	Read manager setup counter
95h	Read value
96h	Write value

Extended Method

Leitura ou Escrita

B1	Endereço de um medidor (<i>Slave Id</i>)
B2	2 (dois)
B3	<i>Handle</i>
B4	Método

Operação de leitura ou escrita utilizando o Método Estendido. Para o endereço do *handle* (parâmetro *B3*), consulte o documento **Module and Register Handles for ION Meters**, disponível com este Driver. Para o método (parâmetro *B4*), consulte a tabela a seguir.

Handles do Método Estendido

CÓDIGO	NOME
03E8h	Read module input handles
03EFh	Read module input names
03ECh	Read module setup handles

CÓDIGO	NOME
0406h	Read module state

Read Log Table

Somente Leitura

B1	Endereço de um medidor (<i>Slave Id</i>)
B2	3 (três)
B3	<i>Handle</i>
B4	Faixa de variação

Para esta leitura, este Driver utiliza um *handle* e um método específicos para cada tabela, configurados através do Tag Index and Range Definitions for Log Table.

Read Log Table - Update New Records First

Somente Leitura

B1	Endereço de um medidor (<i>Slave Id</i>)
B2	7 (sete)
B3	<i>Handle</i>
B4	Faixa de variação

Para esta leitura, este Driver utiliza um *handle* e um método específicos para cada tabela, configurados através do Tag Index and Range Definitions for Log Table. Com esta leitura, este Driver envia uma lista de valores solicitados, mais qualquer novo registro adicionado enquanto este Driver não atinge o ponteiro da tabela.

Read Log Events with Description

Somente Leitura

B1	Endereço do medidor (<i>Slave Id</i>)
B2	8 (oito)
B3	<i>Handle</i>
B4	Faixa de variação

Adiciona 2 (dois) novos Elementos a um Tag Bloco, além dos Elementos originais **Log Position**, **Priority**, **Replace**, **ION Cause**, **Cause Value**, **ION Effect**, **Effect Value** e **Ack Time**. Os novos Elementos são adicionados ao final de um Tag Bloco com uma descrição textual dos *handles* de causa e efeito.

Exemplo de Leitura de Dados de um Energy Recorder

Cada *Energy Data Recorder* possui um *handle*, uma espécie de ponteiro, que aponta para a estrutura. Cada medidor possui um número limitado de *Data Recorders* que podem armazenar qualquer variável, tais como tensão, corrente ou energia, entre outras. Neste exemplo, vamos assumir que o *Data Recorder 1* contém os valores de energia necessários. O próximo passo é criar um Tag Bloco mapeando cada coluna do registro de dados para um Elemento. Neste exemplo, para um medidor 8500, o *handle* para o *Data Recorder 1* é F80h. Para iniciar a leitura, realize os procedimentos a seguir.

1. Associe uma estrutura interna à tabela a ser lida e inicialize esta estrutura.
2. Descubra a posição de log atual.
3. Defina o registro inicial para leitura.
4. Habilite a varredura do Tag Bloco.

Esta aplicação deve executar os passos a seguir, por script.

1. Escrever o valor F80h no Tag X.4.2.0. Esta escrita associa a estrutura zero (parâmetro *N4* igual a zero) ao *handle* F80h.
2. Escrever o valor 28h no Tag X.4.3.0, valor padrão para *Table Pointer Method*.
3. Escrever o valor 10 no Tag X.4.5.0 para trazer, neste exemplo, 10 registros a cada operação de leitura ou varredura.
4. Ler o Tag X.4.4.0, que retorna a posição atual do log.

Assumindo por exemplo que o *Data Register 1* grave registros novos a cada 5 (cinco) minutos, é necessário calcular o número de registros a serem lidos, como nos exemplos da tabela a seguir.

Exemplos de registros

PONTEIRO	HORÁRIO
1000	01/01/2010 00:00
1001	01/01/2010 00:05
1002	01/01/2010 00:10
1003	01/01/2010 00:15
1004	01/01/2010 00:20
1005	01/01/2010 00:25
...	...
1400	02/01/2010 09:20
1401	02/01/2010 09:25
1402	02/01/2010 09:30
1403	02/01/2010 09:35
1404	02/01/2010 09:40

Ler o valor do Tag X.4.4.0 retorna a posição atual do ponteiro. Neste exemplo corresponde à posição 1404, que é o último registro do medidor, 02/01/2010 09:40.

Para ler o dia anterior, deve-se calcular quantos registros retornam. Entre o último registro do medidor, 02/01/2010 09:40, e a data inicial, 01/01/2010 00:00, há 2020 minutos, ou seja, 1 (um) dia, 9 (nove) horas e 40 minutos. A cada 5 (cinco) minutos tem-se 1 (um) registro, portanto deve-se retornar 404 registros. Logo, para ler do medidor registros a partir de 01/01/2010 00:00, deve-se escrever no Tag X.4.0.0 o valor 1000. Após calcular a posição do registro inicial, execute os passos a seguir.

1. Escreva o valor calculado, 1000 para este exemplo, da posição do registro inicial no Tag X.4.0.0.
2. Habilite a varredura do Tag Bloco X.3.F80h.0.

Durante esta varredura, a cada vez que um novo Bloco é coletado de um medidor, são retornados 10 registros e o valor do registro inicial retornado pelo Tag X.4.0.0 é incrementado em 10 unidades, conforme o valor escrito no Tag X.4.5.0. A aplicação de exemplo está disponível no artigo Aplicação Exemplo - Driver Power Measurement ION do **Elipse Knowledgebase**.

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **ION32**.

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
2.0.6	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 38007</i>).
2.0.5	13/09/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (<i>Case 27159</i>).
2.0.3	31/08/2015	F. Englert	• Driver portado para a biblioteca IOKit versão 2 .
2.0.1	21/05/2015	F. Englert	• Corrigido o envio de senhas nos casos em que uma senha era maior do que 6 (seis) caracteres, se configurada na janela de configuração ou usando o parâmetro ION.UserPwd .
1.9.1	28/06/2012	A. Quites	• Corrigida a descrição da opção Master Address no Manual do Usuário (<i>Case 13082</i>). • Este Driver agora é protegido com identificador para aplicações de telemedição (<i>Case 12594</i>).
1.8.1	16/06/2010	A. Quites	• Traduzida a documentação para o português e adicionados exemplos em VBScript (<i>Case 10503</i>). • Corrigido um erro na recepção de <i>frames</i> fora de ordem, que gerava intervalos sem dados na leitura de memória de massa (<i>Case 11167</i>).
1.7.1	10/12/2008	M. Salvador	• Adicionada uma opção para senhas alfanuméricas (<i>Case 8109</i>). • Geração de logs mais detalhados (<i>Cases 8009 e 8020</i>). • Driver portado para a versão mais recente da biblioteca IOKit (<i>Case 7557</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.6.1	06/02/2004	M. Salvador	• Adicionado suporte à comunicação com múltiplos formatos de onda (<i>waveforms</i>).
1.5.1	11/12/2003	M. Salvador	• Adicionados novos Tags com o parâmetro <i>N2</i> igual a 10 e 11, para formatos de onda (<i>waveforms</i>). • Adicionada uma opção para descartar bytes FF duplicados para uso em canais de comunicação ruidosos, geralmente em <i>links</i> seriais.
1.3.1	06/10/2003	M. Salvador	• Este Driver agora rejeita pacotes com o endereço de origem igual ao endereço deste Driver, ou seja, pacotes gerados por ecos de pacotes enviados pelo próprio Driver (<i>Case 2647</i>).
1.2.1	30/09/2003	F. Englert	• Adicionado suporte a endereços de medidores entre 0 (zero) e 65535. Versões anteriores suportavam medidores entre 0 (zero) e 255 (<i>Case 2633</i>).
1.0.1	17/12/2002	F. Englert	• Versão inicial deste Driver portada para a biblioteca IOKit.