

## DRIVER DA CAD12/36 PARA WINDOWS NT 4.0

### PARA USO COM O LABVIEW

#### 1. INTRODUÇÃO

---

##### 1.1. Escopo

O objetivo deste documento é apresentar as funções da biblioteca DLL de acesso à placa CAD12/36 para uso com o LabVIEW.

##### 1.2. Audiência

Esta documentação se destina aos programadores de aplicativos em LabVIEW 5.0 que realizam aquisição de sinais com a placa CAD12/36.

Para um bom entendimento desta nota de aplicação, recomenda-se que o leitor tenha conhecimento prévio dos seguintes assuntos:

- aquisição de dados
- utilização da CAD12/36
- LabVIEW 5.0 for Windows 95

##### 1.3. Aplicabilidade

O driver NT4\_1236 deve ser utilizado somente no sistema operacional Windows NT 4.0 ou compatível e com a placa A/D CAD12/36 (exceto a revisão R4). O suporte a saídas analógicas é disponível para as opções D/A: 2x12 bits, 4x12 bits, 1x14 bits (configuração por jumper) e 1x15 bits.

Para maiores informações sobre a CAD12/36, consulte o manual de referência técnica da placa.

---

	Elaboração	Aprovação
Área	SW.P&D	SW.P&D
Nome	Lauro	Lauro
Data	15/dez/1999	15/dez/1999
Visto		

Reprodução Proibida

## 2. INSTALANDO O DRIVER NT4\_1236

---

Os arquivos fornecidos para utilização do driver NT4\_1236:

<b>NT4_1236.SYS</b>	kernel mode driver para Windows NT 4.0
<b>InstDrv.exe</b>	programa de instalação do driver no registry do Windows NT 4.0
<b>RegIni.exe</b>	programa de registro dos parâmetros do driver no registry do Windows NT 4.0
<b>NT4_1236.ini</b>	arquivo de registro do driver
<b>NT4_1236.DLL</b>	biblioteca DLL de acesso ao driver
<b>NLV_1236.DLL</b>	biblioteca DLL de acesso ao driver para uso com o LabVIEW
<b>NT4api.pas</b>	módulo em pascal para a interface com o driver
<b>TESTE.EXE</b>	programa de teste do driver
<b>Delphi\*.*</b>	exemplo com programa fonte em Borland Delphi
<b>VB5\*.*</b>	exemplo com programa fonte em Visual Basic 5.0
<b>Read_AI.VI</b>	exemplo em LabVIEW com aquisição por pooling
<b>Acq_Int.VI</b>	exemplo em LabVIEW com aquisição por interrupção

### 2.1. Instalando o Driver

Siga as seguintes etapas para instalar o driver NT4\_1236:

1. A partir da barra de tarefa, execute **Iniciar/Programas/Prompt de comando** para entrar na janela *Prompt de comando* (Janela DOS).
2. Supondo que o Windows NT esteja instalado no diretório C:\WINNT, execute o seguinte comando para copiar o driver para o diretório do Windows.

```
COPY A:\NT4_1236.SYS C:\WINNT\SYSTEM32\DRIVERS
```

3. Mude o diretório corrente para A:\:

```
CD A:\  
A:
```

4. Execute o seguinte comando para registrar a configuração padrão do driver no *Registry do Windows NT*:

```
REGINI NT4_1236.INI
```

5. Execute o seguinte comando para instalar o driver no *Registry do Windows NT*:

```
INSTDRV NT4_1236 C:\WINNT\SYSTEM32\DRIVERS\NT4_1236.SYS
```

### 2.2. Como Alterar a Configuração Padrão do Driver

A configuração padrão do driver *NT\_1236* pode ser alterada através do *Registry do Windows NT*. Os parâmetros configuráveis se referem a:

- ◆ Endereço base de I/O da CAD12/36
- ◆ Canal de interrupção utilizado
- ◆ Canal de DMA utilizado (reservado para uso futuro)

Para modificar esses parâmetros, você deverá utilizar o programa *Editor do Registro* do Windows NT. Esse programa se encontra no sub-diretório SYSTEM32 do diretório de instalação do Windows NT. O nome do arquivo executável do *Editor do Registro* é REGEDT32.EXE.

Após entrar no *Editor do Registro*, selecione a janela **HKEY\_LOCAL\_MACHINE** e abra a pasta

**SYSTEM\CurrentControlSet\Services\NT4\_1236\Parameters**. Esta pasta possui os seguintes valores correspondentes aos parâmetros de configuração do driver:

Nome do Parâmetro	Parâmetro	Valor padrão
IoBaseAddr	Endereço base de I/O da placa CAD12/36	0x380
Interrupt	Canal de interrupção utilizado pela CAD12/36	5
DMA	reservado para uso futuro	1

Para consultar os valores de endereços de I/O que estão sendo utilizados por outros dispositivos instalados no Windows NT, utilize o programa *Diagnóstico do Windows NT*. Para isso execute a partir da barra de tarefas **Iniciar/Ferramentas administrativas/Diagnóstico do Windows NT**. No *Diagnóstico do Windows NT*, consulte a página *Recursos*.

Para informações mais detalhadas dos programas *Editor de Registro* e *Diagnóstico do Windows NT*, consulte documentação da Microsoft.

Após modificar os parâmetros do driver, você deverá reiniciar o Windows NT para que os novos valores passem a ter efeito. Ou se preferir, dê um duplo clique sobre o ícone *Dispositivos* no *Painel de Controle*. Na janela do programa *Dispositivos*, selecione a linha correspondente ao *NT4\_1236* e clique sobre o botão *Finalizar* e em seguida sobre o botão *Iniciar*, para reiniciar o driver com os novos parâmetros.

### 3. INTERFACE COM O LABVIEW

---

O acesso do programa em LabVIEW às funções do driver da CAD12/36 é realizada através do bloco *Call Library Function* do LabVIEW. Para cada chamada de uma função do driver deve haver um correspondente bloco *Call Library Function*.

Na configuração do *Call Library Function*, deve-se especificar no campo *Library Name or Path* o valor **NLV\_1236.DLL** e no campo *Calling Conventions* o valor **C**. Os demais campos da configuração são dependentes da função. O próximo tópico descreve as funções da biblioteca DLL de acesso ao driver da CAD12/36 para Windows NT 4.0.

#### 3.1. Estrutura de Dados do Driver

O driver mantém uma estrutura de dados interna com todas as informações necessárias para a utilização da CAD12/36. Algumas dessas informações (endereço base padrão, canal de interrupção e DMA) são passadas para o driver através do Registry do Windows NT.

O driver possui uma memória de canais onde são armazenados a ordem em que os canais serão adquiridos e os seus respectivos ganhos. Com essa estrutura você pode, por exemplo, informar ao driver para adquirir 4 canais na seguinte ordem e ganhos: canal 0 em  $\pm 5$  volts, canal 3 em  $\pm 10$  volts, canal 12 em  $\pm 2.0$  volts e o canal 15 em 0-10 volts).

O driver possui um buffer circular de 64K amostras para a aquisição de sinais. Por exemplo, para a aquisição de 4 canais a 1000 Hz por canal, o buffer teria capacidade de armazenar até 16 segundos sem que o programa aplicativo remova os dados do buffer circular.

#### 3.2. Constantes

A tabela seguinte apresenta as constantes retornadas pela primitiva *NT4\_GetAcquiredData*.

Constante	Valor	Descrição
ieNoError	0	Não ocorreu nenhum erro desde a última chamada da primitiva <i>NT4_GetAcquiredData</i>
ieIntOverrun	1	Indica que houve interrupt overrun. Ou seja, antes do término do tratamento da interrupção, houve um novo pedido de interrupção da placa A/D.
ieAD_Error	2	Indica que houve erro no conversor A/D da placa.
ieBufOverflow	3	Overflow no buffer de aquisição do driver
ieDriver	100	Erro de acesso ao driver
ieAcqStop	101	Aquisição não está ativa
ieLabView	102	Dimensão errada no buffer do LabView passado na função <i>NT4_GetSamples</i> . Número de colunas é diferente do número de sinais habilitados para aquisição.

A tabela seguinte apresenta as constantes retornadas pela primitiva *NT4\_AcquisitionSetup*.

Constante	Valor	Descrição
seNoError	0	Nenhum erro
seNoSignal	1	Não foi programado nenhum sinal para a aquisição.

## 4. DESCRIÇÃO DAS FUNÇÕES DO NLV\_1236.DLL

Neste tópico são descritas as funções da biblioteca DLL de acesso ao driver da CAD12/36 especialmente implementada para interfaceamento com o LabVIEW. A tabela abaixo lista as funções do NLV\_1236.DLL.

Primitiva	Descrição
NT4_Open	Inicia o acesso ao driver
NT4_Close	Finaliza o acesso ao driver
NT4_GetVersion	Informa a versão e revisão do driver
NT4_GetAIO_Caps	Informa os recursos de entrada e saída analógica disponíveis no driver
NT4_GetDIO_Caps	Informa os recursos de entrada e saída digital disponíveis no driver
NT4_SetAiRange	Seleciona faixa de entrada do conversor A/D
NT4_ReadAi	Leitura de canal de entrada analógica
NT4_WriteAo	Escreve em canal de saída analógica
NT4_ReadDI	Leitura dos ports de entrada digital P0 e P1
NT4_WriteDO	Escrita nos ports de saída digital P0 e P1
NT4_Clear_CM	Limpa a memória de canais da aquisição por interrupção
NT4_Insert_AI_CM	Inserir um canal de entrada analógica para aquisição
NT4_Insert_DI_CM	Inserir os ports de saída digital para aquisição (P0 e P1)
NT4_AcquisitionSetup	Programa aquisição por interrupção
NT4_StartAcquisition	Inicia a aquisição por interrupção
NT4_StopAcquisition	Finaliza a aquisição por interrupção
NT4_GetSamples	Leitura das amostras do buffer de aquisição por interrupção

### 4.1. Primitiva NT4\_Open

```
uInt8 NT4_Open (void);
```

Esta primitiva abre o acesso ao driver e deve ser a primeira primitiva a ser executada pelo programa aplicativo. A primitiva retorna o valor *zero* quando a primitiva foi executada com sucesso. Uma das causas possíveis são:

Causa	Verificação	Solução
Driver não instalado	No Painel de Controle do Windows NT, clique sobre Dispositivos. Se o driver NT4_1236 estiver instalado, ele deve estar na lista de dispositivos apresentada.	Instale o driver
Driver não iniciado ou erro na parametrização	No Painel de Controle do Windows NT, clique sobre Dispositivos. Verifique se o driver foi iniciado.	Proceda como descrito em 2.2.
Placa não instalada ou falha na iniciação da placa	Verifique se o endereço base da placa e o canal de interrupção configurado por jumpers na placa estão coerentes com o especificado na parametrização do driver. Para mudar a parametrização do driver, consulte 2.2.	Instale a placa CAD12/36 e/ou altere a parametrização do driver.

Veja também a primitiva *NT4\_Close*.

## 4.2. Primitiva NT4\_Close

```
void NT4_Close (void);
```

Esta primitiva finaliza o acesso ao driver e deve ser a última primitiva a ser executada pelo programa aplicativo. Para cada chamada bem sucedida da primitiva *NT4\_Open* deve haver uma correspondente chamada da *NT4\_Close*.

Veja também a primitiva *NT4\_Open*.

## 4.3. Primitiva NT4\_GetVersion

```
uInt8 NT4_GetVersion (uInt8 *VersionHigh, uInt8 *VersionLow);
```

Estando o driver instalado, esta primitiva retorna 1 (*true*) e devolve nos parâmetros *VersionHigh* e *VersionLow*, respectivamente o byte mais significativo e o byte menos significativo da versão do device driver.

## 4.4. Primitiva NT4\_GetAIO\_Caps

```
uInt8 NT4_GetAIO_Caps (uInt16 *nAiChannels, uInt16 *nAoChannels,  
                      uInt16 *nAiRange, Array1DSingle **AiRange);
```

Estando o device driver instalado e operando, esta primitiva retorna (1) *true* e informa nos parâmetros de saída os recursos de entrada e saída analógica disponibilizados pelo device driver.

Nos parâmetros *nAiChannels* e *nAoChannels* são retornados respectivamente, o número de entradas analógicas e o número de saídas analógicas da placa.

O valor retornado em *nAiRange* indica o número de faixas de entrada do conversor A/D. No parâmetro *AiRange* é retornado um vetor de 16 posições (apenas 6 posições são usadas) com os valores das faixas de entrada do conversor A/D em volts. Valores negativos indicam faixa de entrada bipolar, por exemplo o valor -10.0 indica que a faixa de entrada correspondente é de -10 a 10 volts. Os valores positivos indicam que a faixa de entrada é unipolar, por exemplo, o valor 5.0 indica que a faixa de entrada é de 0 a 5 volts. O conteúdo do vetor é apenas informativo para o programa aplicativo. Para selecionar uma faixa de entrada, o programa deverá informar o índice correspondente ao vetor. Para a placa CAD12/36 são disponíveis as faixas de entrada do A/D apresentadas na tabela.

Índice	Valor do AiRange	Faixa de Entrada
0	-10.0	-10.0 a 10.0 volts
1	-5.0	-5.0 a 5.0 volts
2	-2.0	-2.0 a 2.0 volts
3	5.0	0 a 10.0 volts
4	5.0	0 a 5.0 volts
5	2.0	0 a 2.0 volts

Veja também a primitiva *NT4\_GetDIO\_Caps*.

#### 4.5. Primitiva NT4\_GetDIO\_Caps

```
uInt8 NT4_GetDIO_Caps (uInt16 *nDiPorts, uInt16 *nBitsDI,  
                      uInt16 *nDoPorts, uInt16 *nBitsDO);
```

Estando o device driver instalado e operando, esta primitiva retorna 1 (*true*) e informa nos parâmetros de saída os recursos de entrada e saída digital disponibilizados pelo device driver.

O número de ports de entrada digital é retornado no parâmetro *nDiPorts* e o número de pontos por port de entrada digital é retornado em *nBitsDI*.

O número de ports de saída digital é retornado no parâmetro *nDoPorts* e o número de pontos por port de saída digital é retornado em *nBitsDO*.

Veja também a primitiva *NT4\_GetAIO\_Caps*.

#### 4.6. Primitiva NT4\_SetAiRange

```
uInt8 NT4_SetAiRange (uInt8 iRange);
```

Esta primitiva permite selecionar a faixa de entrada da placa A/D. Ao ser iniciada a CAD12/36 o driver programa o ganho para a faixa de entrada de  $\pm 10$  volts. Se desejar uma outra faixa de entrada, o aplicativo deve chamar esta primitiva passando no parâmetro *iRange* o número da faixa de entrada desejada (veja 4.4. Primitiva *NT4\_GetAIO\_Caps*). A primitiva retorna 1 (*true*) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são faixa de entrada inválida e driver não instalado.

Veja também as primitivas *NT4\_ReadAi* e *NT4\_GetAIO\_Caps*.

#### 4.7. Primitiva NT4\_ReadAi

```
uInt8 NT4_ReadAi (uInt8 Channel, Int16 *Value);
```

Esta primitiva realiza a leitura de um canal de entrada analógica da CAD12/36. O programa aplicativo deve passar no parâmetro *Channel* o número do canal A/D a ser lido. A faixa de entrada a ser utilizada na conversão do canal A/D deve ser previamente programada através da primitiva *NT4\_SetAiRange*, caso seja uma faixa diferente da última faixa programada.

No parâmetro de saída *Value* é retornado o valor lido do conversor A/D. O valor lido é representado em complemento de 2 e pode assumir valores de -32768 a 32767. A primitiva retorna 1 (*true*) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são canal de entrada analógica inválido, erro no conversor A/D e driver não instalado.

Esta primitiva não pode ser chamada durante a aquisição por interrupção.

Veja também as primitivas *NT4\_SetAiRange* e *NT4\_GetAIO\_Caps*.

#### 4.8. Primitiva NT4\_WriteAo

```
uInt8 NT4_WriteAo (uInt8 Channel, Int16 Value);
```

Esta primitiva realiza a escrita em canal de saída analógica da CAD12/36. O programa aplicativo deve passar no parâmetro *Channel* o número do canal D/A a ser atualizado. O valor a ser escrito no D/A deve ser passado no parâmetro *Value*, que deve estar representado em complemento de 2 com valores de -32768 a 32767. Esses valores correspondem, respectivamente, aos limites inferior e superior da saída analógica. A

primitiva retorna 1 (true) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são canal de saída analógica inválido e driver não instalado.

Esta primitiva não pode ser chamada durante a aquisição por interrupção.

Veja também a primitiva *NT4\_GetAIO\_Caps*.

#### 4.9. Primitiva NT4\_ReadDI

```
uInt8 NT4_ReadDI (uInt8 Group, uInt16 *Value);
```

Esta primitiva realiza a leitura de um port de entrada digital da CAD12/36. O programa aplicativo deve passar no parâmetro *Group* o número do port de entrada digital a ser lido. O número de ports de entrada digital disponíveis na placa pode ser consultado através da primitiva *NT4\_GetDIO\_Caps*. No caso da CAD12/36 é disponível 1 port de entrada digital de 16 bits.

No parâmetro de saída *Value* é retornado o valor lido do port de entrada digital. A primitiva retorna 0 (zero) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são número do port de entrada digital inválido e driver não instalado.

Veja também as primitivas *NT4\_WriteDO* e *NT4\_GetDIO\_Caps*.

#### 4.10. Primitiva NT4\_WriteDO

```
uInt8 NT4_WriteDO (uInt8 Group, uInt16 Value);
```

Esta primitiva realiza a escrita em port de saída digital da CAD12/36. O programa aplicativo deve passar no parâmetro *Group* o número do port de saída digital a ser atualizado. O número de ports de saída digital disponíveis na placa pode ser consultado através da primitiva *NT4\_GetDIO\_Caps*. No caso da CAD12/36 é disponível 1 port de saída digital de 16 bits.

O valor a ser escrito no port de saída digital deve ser passado no parâmetro *Value*. A primitiva retorna 1 (true) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são número do port de saída digital inválido e driver não instalado.

Veja também as primitivas *NT4\_ReadDI* e *NT4\_GetDIO\_Caps*.

#### 4.11. Primitiva NT4\_Clear\_CM

```
void NT4_Clear_CM (void);
```

Esta primitiva limpa a memória de canais utilizada na aquisição por interrupção. Através da memória de canais, o aplicativo informa ao driver a relação dos canais a serem adquiridos durante a aquisição por interrupção. Para cada canal analógico a ser adquirido, o aplicativo deverá realizar uma chamada da primitiva *NT4\_Insert\_AI\_CM*. Analogamente, deve-se executar a primitiva *NT4\_Insert\_DI\_CM* para os ports de entrada digital. Antes, porém, o aplicativo deverá limpar a memória de canais através da chamada da primitiva *NT4\_Clear\_CM*.

Veja também as primitivas *NT4\_Insert\_AI\_CM* e *NT4\_Insert\_DI\_CM*.



#### 4.12. Primitiva NT4\_Insert\_AI\_CM

```
uInt8 NT4_Insert_AI_CM (uInt8 Channel, uInt8 iRange);
```

O programa aplicativo deve realizar chamadas sucessivas desta primitiva para informar os canais de entrada analógica a serem lidos na aquisição por interrupção. A ordem de chamada desta primitiva determina a ordem em que os canais serão lidos. Antes da chamada desta primitiva para especificar o primeiro canal a ser adquirido, deve-se chamar a primitiva *NT4\_Clear\_CM* para limpar a memória de canais.

Os parâmetros de entrada *Channel* e *iRange* correspondem respectivamente ao número do canal A/D e o índice da faixa de entrada (veja 4.4. Primitiva *NT4\_GetAIO\_Caps*). A primitiva retorna 1 (true) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são canal de entrada analógica inválido, faixa de entrada inválida, excedeu a capacidade da memória de canais e driver não instalado.

Veja também as primitivas *NT4\_Insert\_DI\_CM* e *NT4\_Clear\_CM*.

#### 4.13. Primitiva NT4\_Insert\_DI\_CM

```
uInt8 NT4_Insert_DI_CM (uInt8 Group);
```

O programa aplicativo deve realizar chamadas sucessivas desta primitiva para informar os ports de entrada digital a serem lidos na aquisição por interrupção. A ordem de chamada desta primitiva determina a ordem em que os ports serão lidos.

O parâmetro *Group* corresponde ao número do port de entrada de entrada digital. Os ports de entrada digital são lidos em grupo de 16 bits e no caso da placa CAD12/36, deve-se passar o valor 0 (zero) neste parâmetro para que o driver leia os ports P0 e P1. A primitiva retorna 0 (zero) se foi executada com sucesso. Os erros mais comuns na execução da primitiva são número do port de entrada digital inválido, excedeu a capacidade da memória de canais e driver não instalado.

Veja também as primitivas *NT4\_Insert\_AI\_CM* e *NT4\_Clear\_CM*.

#### 4.14. Primitiva NT4\_AcquisitionSetup

```
uInt8 NT4_AcquisitionSetup (single SampleFreq, uInt32 *ErrorCode);
```

Esta primitiva prepara a aquisição de sinais por interrupção e deve ser executada depois da programação da memória de canais através das funções *NT4\_Clear\_CM*, *NT4\_Insert\_AI\_CM* e *NT4\_Insert\_DI\_CM*.

A frequência em que os sinais serão amostrados deve ser especificada em hertz no parâmetro *SampleFreq*. No parâmetro de saída *ErrorCode* é retornado o código de erro na programação da aquisição (veja 3.2 Constantes). A primitiva retorna 1 (true) se foi executada com sucesso.

Veja também as primitivas *NT4\_Clear\_CM*, *NT4\_Insert\_AI\_CM*, *NT4\_Insert\_DI\_CM*, *NT4\_StartAcquisition*, *NT4\_GetSamples* e *NT4\_StopAcquisition*.

#### 4.15. Primitiva NT4\_StartAcquisition

```
void NT4_StartAcquisition (void);
```

Esta primitiva inicia a aquisição de sinais por interrupção com os parâmetros programados anteriormente. Ela deve ser executada depois da programação da memória de canais e da preparação da aquisição através das primitivas *NT4\_Clear\_CM*, *NT4\_Insert\_AI\_CM*, *NT4\_Insert\_DI\_CM* e *NT4\_AcquisitionSetup*.

Após a chamada da *NT4\_StartAcquisition*, o programa aplicativo tem acesso ao andamento da aquisição através da primitiva *NT4\_GetSamples*.

Veja também as primitivas *NT4\_Clear\_CM*, *NT4\_Insert\_AI\_CM*, *NT4\_Insert\_DI\_CM*, *NT4\_AcquisitionSetup*, *NT4\_GetSamples* e *NT4\_StopAcquisition*.

#### 4.16. Primitiva *NT4\_StopAcquisition*

```
void NT4_StopAcquisition (void);
```

Esta primitiva encerra a aquisição de sinais por interrupção.

Veja também a primitiva *NT4\_StartAcquisition*.

#### 4.17. Primitiva *NT4\_GetSamples*

```
uInt8 NT4_GetSamples (uInt32 *ieStatus, uInt32 *iSample, uInt32 *nSampRead,  
                    Array2DInt16 *Samples);
```

Através desta função o programa aplicativo tem acesso às amostras aqisitadas pelo driver durante a aquisição de sinais por interrupção.

O driver possui um buffer circular de 64K amostras para a aquisição de sinais. Por exemplo, para a aquisição de 16 canais a 140 Hz por canal, o buffer teria capacidade de armazenar até 29 segundos sem que o programa aplicativo remova os dados do buffer circular.

No entanto, o programa aplicativo normalmente possui um loop de processamento onde é realizada periodicamente a chamada da função *NT4\_GetSamples* para a leitura dos dados amostrados. O tamanho do buffer de aquisição do driver é utilizado apenas para que não haja perdas de dados durante processamentos demorados do programa aplicativo. No exemplo, o programa aplicativo pode ficar até 29 segundos sem retirar dados do buffer de aquisição. Depois deste tempo ocorre *overflow* no buffer.

A função retorna no parâmetro de saída *ieStatus* o código de ocorrência de erro durante a aquisição (veja 3.2 Constantes).

O parâmetro de saída *Samples* é um array do LabVIEW com duas dimensões. A primeira dimensão representa o índice da amostragem e a segunda dimensão o índice do canal (ordem na memória de canais). Ou seja, *Samples* é uma tabela em que as linhas são os “instantes” de amostragem e as colunas os canais.

A função remove as amostras do buffer de aquisição do driver e as transfere para o *Samples*. O número de amostras transferidas para o array é limitado pelo tamanho do array e pelo número de amostras disponíveis no buffer de aquisição do driver. O número de amostras por canal transferidos para o array é retornado no parâmetro de saída *nSampRead*. No parâmetro *iSample* é retornado o número de ordem da última amostra transferida para o array. O número de ordem da amostra se refere ao início da aquisição. Em cada chamada da função *NT4\_GetSamples* é transferido no máximo um total de 16K amostras.

A função retorna o valor 1 (true) se foi transferido dados para o array.

Veja também a função *NT4\_StartAcquisition* e *NT4\_StopAcquisition*.