

# DRIVER DO ADS2000

## PARA

### LABVIEW FOR WINDOWS 95

## 1. INTRODUÇÃO

### 1.1. Escopo

O objetivo deste documento é apresentar as funções da biblioteca DLL de acesso ao ADS2000 para uso com o LabVIEW.

### 1.2. Audiência

Esta documentação se destina aos programadores de programas aplicativos em LabVIEW 5.0 for Windows 95 que realizam aquisição de sinais com o módulo de aquisição de dados ADS2000.

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

- aquisição de dados
- utilização do ADS2000
- LabVIEW 5.0 for Windows 95

### 1.3. Aplicabilidade

O driver VADSD deve ser utilizado somente no sistema operacional Windows 95 ou compatível. O módulo ADS2000 deverá conter um cartão de controle AC-2120 e 1 ou 2 módulos condicionadores de sinais. Os endereços dos módulos condicionadores devem ser setados para 32 e 64. O driver determina o endereço dos cartões de condicionamento através do número do canal. Assim, os canais 0 a 15 devem estar no cartão de endereço 32 e os canais 16 a 31 no cartão de endereço 64 no barramento ADS.

Para maiores informações sobre o ADS2000, consulte o manual de referência técnica do ADS2000.

	Elaboração	Aprovação
Área	SW.P&D	SW.P&D
Nome	Lauro	Lauro
Data	03/jun/1999	03/jun/1999
Visto		

**Reprodução Proibida**

## 2. INSTALANDO O DRIVER VADSD

---

Os arquivos fornecidos para utilização do driver VADSD são:

<b>VADSD.VXD</b>	driver do AC2120 / ADS2000 para Windows 95
<b>VADSD.DLL</b>	biblioteca DLL que faz a interface com o driver VADSD
<b>TESTE.EXE</b>	programa de teste do driver / ADS2000
<b>Read_AI.VI</b>	programa em LabVIEW usando o VADSD com aquisição por pooling
<b>Acq_Int.VI</b>	programa em LabVIEW usando o VADSD com aquisição por interrupção

A instalação do driver do ADS2000 é bastante simples, basta copiar os arquivos VADSD.VXD e VADSD.DLL para o diretório onde se encontra o programa aplicativo em LabVIEW.

O driver VxD do ADS2000 é carregado em tempo de execução do programa aplicativo, não sendo necessário nenhum registro no Windows 95 através do REGEDIT.

Como a interface do ADS2000 com o microcomputador é realizada através da interface paralela do microcomputador (por exemplo a LPT1), o driver requisita um acesso exclusivo da porta paralela durante o uso do ADS2000. Por isso, outros dispositivos conectados na interface paralela (impressora, scanner, ZIP driver, CD-ROM, etc) não poderão ser utilizados durante a aquisição de sinais com o ADS2000.

O programa TESTE.EXE é um programa de teste fornecido com o driver, que permite testar o ADS2000 e o respectivo driver.

Os arquivos Read\_AI.VI e Acq\_Int.VI são programas exemplos em LabVIEW.

## 3. INTERFACE COM O LABVIEW

---

O acesso do programa em LabVIEW às funções do driver do ADS2000 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 **VADSD.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 ADS2000.

## 4. DESCRIÇÃO DAS FUNÇÕES DO VADSD.DLL

---

Neste tópico são descritas as funções da biblioteca DLL de acesso ao ADS2000 especialmente implementada para interfaceamento com o LabVIEW. A tabela abaixo lista as funções do VADSD.DLL.

Função	Descrição
VADSD_Open	Inicia o acesso ao driver do ADS2000
VADSD_Close	Finaliza o acesso ao driver do ADS2000
VADSD_OpenCard	Inicia o ADS2000
VADSD_CloseCard	Finaliza o acesso ao ADS2000
VADSD_ReadAI	Leitura de canal de entrada analógica
VADSD_ReadDI	Leitura de módulo de entrada digital no barramento ADS
VADSD_WriteDO	Escrita em módulo de saída digital no barramento ADS
VADSD_ReadDO	Leitura do último valor escrito em módulo de saída digital no barramento ADS
VADSD_ClearCM	Limpa a memória de canais da aquisição por interrupção
VADSD_WriteCM	Escreve na memória de canais da aquisição por interrupção
VADSD_StartAcquisition	Inicia a aquisição por interrupção
VADSD_StopAcquisition	Finaliza a aquisição por interrupção
VADSD_GetSamples	Leitura das amostras do buffer de aquisição por interrupção

### 4.1. Função VADSD\_Open

```
uInt8 VADSD_Open (void);
```

Esta função abre o acesso ao driver do ADS2000 e deve ser a primeira função da biblioteca DLL a ser executada pelo programa aplicativo. A função retorna o valor 1 quando a função foi executada com sucesso ou o valor 0 (zero) quando houve algum erro na execução da função. Normalmente o erro ocorre quando o driver não foi encontrado. Neste caso, verifique se o arquivo **VADSD.VXD** se encontra no diretório onde o programa está sendo executado.

Veja também a função *VADSD\_Close*.

### 4.2. Função VADSD\_Close

```
void VADSD_Close (void);
```

Esta função finaliza o acesso ao driver do ADS2000 e deve ser a última função da biblioteca DLL a ser executada pelo programa aplicativo. Para cada chamada bem sucedida da função *VADSD\_Open* deve haver uma correspondente chamada da *VADSD\_Close*.

Veja também a função *VADSD\_Open*.

### 4.3. Função VADSD\_OpenCard

```
uInt8 VADSD_OpenCard (uInt8 LPTn, uInt16 BaseAddr, uInt8 IRQ);
```

Esta função inicia o ADS2000 e permite o acesso às demais funções da biblioteca DLL.

O programa em LabVIEW deve passar no parâmetro *LPTn* o número da porta paralela em que o ADS2000 está conectado. Por exemplo, caso se esteja utilizando a interface paralela LPT1, deve-se passar o valor 1 neste parâmetro.

Deve-se passar no parâmetro *BaseAddr* o endereço base de I/O da porta paralela. Normalmente o endereço base de I/O 378h é utilizado pela LPT1 e o endereço 278h pela LPT2.

Caso a aplicação em LabVIEW for utilizar aquisição por interrupção, deve-se especificar no parâmetro *IRQ* o canal de interrupção utilizada pela interface paralela. O valor padrão do canal de interrupção é 7 para a LPT1 e 5 para a LPT2.

Os valores exemplificados nesta documentação são os valores usualmente utilizados nos microcomputadores, no entanto alguns fabricantes de micros podem utilizar valores diferentes. Consulte na documentação do seu microcomputador os valores adequados para o seu microcomputador.

A função *VADSD\_OpenCard* retorna o valor 1 se a iniciação do ADS2000 foi bem sucedida ou o valor 0 (zero) se ocorreu algum erro durante a sua execução. Em caso de erro, verifique se o número da porta paralela e o endereço base estão corretos e se o ADS2000 se encontra ligado e conectado à interface paralela. Se o erro persistir, verifique se não existe nenhum outro dispositivo ou driver que esteja utilizando exclusivamente a interface paralela, por exemplo: scanner, câmera de video conferência, ZIP driver, driver de CD-ROM, etc.

Se a função for realizada com sucesso, a interface paralela passa a ser de uso exclusivo do driver até a sua liberação através da chamada da função *VADSD\_CloseCard*.

Veja também a função *VADSD\_CloseCard*.

#### 4.4. Função *VADSD\_CloseCard*

```
uInt8 VADSD_CloseCard (void);
```

Esta função finaliza o acesso ao ADS2000 e libera o uso exclusivo da interface paralela. Ela deve ser chamada antes da função *VADSD\_Close*. Para cada chamada bem sucedida da função *VADSD\_OpenCard* deve haver uma correspondente chamada da *VADSD\_CloseCard*.

Veja também a função *VADSD\_OpenCard*.

#### 4.5. Função *VADSD\_ReadAI*

```
uInt8 VADSD_ReadAI (uInt8 Channel, uInt8 GainCode, Int16 *Value);
```

Esta função realiza a leitura de um canal analógico do ADS2000.

O programa em LabVIEW deve passar no parâmetro *Channel* o número do canal A/D a ser lido. Os canais 0 a 15 correspondem aos canais do módulo de condicionamento de sinais de endereço 32 no barramento ADS e os canais 16 a 31 correspondem aos canais do módulo de endereço 64.

GainCode	Faixa de Entrada
0	0 a 10 volts
1	+/- 10 volts
2	0 a 5 volts
3	+/- 5 volts
4	0 a 2.5 volts
5	+/- 2.5 volts
6	0 a 1.25 volts
7	+/- 1.25 volts

A faixa de entrada a ser utilizada na conversão do canal A/D deve ser passada no parâmetro *GainCode*. A tabela apresenta os valores válidos para este parâmetro e as respectivas faixas de entrada em volts.

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 função retorna 1 se a conversão foi realizada com sucesso ou 0 se houve algum erro na execução da função.

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

#### 4.6. Função VADSD\_ReadDI

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

Esta função realiza a leitura de um módulo de entrada digital no barramento ADS.

O programa em LabVIEW deve passar no parâmetro *Group* o endereço do módulo no barramento ADS. Os valores válidos para este parâmetro são 0 a 127.

No parâmetro de saída *Value* é retornado o valor lido do módulo de entrada digital.

A função retorna 1 se a leitura foi realizada com sucesso ou 0 se houve algum erro na execução da função.

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

#### 4.7. Função VADSD\_WriteDO

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

Esta função realiza uma escrita em módulo de saída digital no barramento ADS.

O programa em LabVIEW deve passar no parâmetro *Group* o endereço do módulo no barramento ADS. Os valores válidos para este parâmetro são 0 a 127. Deve-se passar no parâmetro *Value* o valor a ser escrito no módulo de saída digital.

A função retorna 1 se a escrita foi realizada com sucesso ou 0 se houve algum erro na execução da função.

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

Veja também a função *VADSD\_ReadDO*.

#### 4.8. Função VADSD\_ReadDO

```
int16 VADSD_ReadDO (uInt8 Group);
```

Esta função retorna o último valor escrito no módulo de saída digital de endereço *Group* no barramento ADS.

Veja também a função *VADSD\_WriteDO*.

#### 4.9. Função `VADSD_ClearCM`

```
void VADSD_ClearCM (void);
```

Esta função limpa a memória de canais utilizada na aquisição por interrupção. Através da memória de canais, o aplicativo informa o driver a relação dos canais a serem aquiridos durante a aquisição por interrupção. Para cada canal a ser aquirido, o aplicativo deverá realizar uma chamada da função `VADSD_WriteCM`. Antes, porém, o aplicativo deverá limpar a memória de canais através da chamada da função `VADSD_ClearCM`.

Veja também a função `VADSD_WriteCM`.

#### 4.10. Função `VADSD_WriteCM`

```
void VADSD_WriteCM (uInt8 Channel, uInt8 GainCode);
```

O programa aplicativo deve realizar chamadas sucessivas desta função para informar os canais a serem lidos na aquisição por interrupção. A ordem de chamada desta função determina a ordem em que os canais serão lidos. Antes da chamada desta função para especificar o primeiro canal a ser aquirido, deve-se chamar a função `VADSD_ClearCM` para limpar a memória de canais.

Os parâmetros de entrada `Channel` e `GainCode` têm o mesmo significado descrito na função `VADSD_ReadAI`.

Veja também a função `VADSD_ClearCM` e `VADSD_ReadAI`.

#### 4.11. Função `VADSD_StartAcquisition`

```
uInt8 VADSD_StartAcquisition (single FreqAcq);
```

Esta função inicia a aquisição de sinais por interrupção. Ela deve ser executada depois da programação da memória de canais através das funções `VADSD_ClearCM` e `VADSD_WriteCM`.

A frequência em que os sinais serão amostrados deve ser especificada em hertz no parâmetro `FreqAcq`.

A função retorna o valor 1 se a execução foi bem sucedida ou valor 0 se houve alguma erro na sua execução. As causas possíveis são: nenhum canal programado para aquisição, driver não iniciado ou ADS2000 não iniciado.

Após a chamada da `VADSD_StartAcquisition`, o programa aplicativo tem acesso aos dados aquiridos através da função `VADSD_GetSamples`.

Veja também a função `VADSD_StopAcquisition` e `VADSD_GetSamples`.

#### 4.12. Função `VADSD_StopAcquisition`

```
void VADSD_StopAcquisition (void);
```

Esta função encerra a aquisição de sinais por interrupção. Para cada chamada bem sucedida da função `VADSD_StartAcquisition` deve haver uma correspondente chamada da `VADSD_StopAcquisition`.

Veja também a função `VADSD_StartAcquisition`.

### 4.13. Função VADSD\_GetSamples

```
uInt8 VADSD_GetSamples (uInt32 *ErrorCode, uInt32 *iSample,  
                        uInt32 *nSampRead, Array2DInt16 *Samples);
```

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

O driver possui um buffer circular de 256K amostras para a aquisição de sinais. Por exemplo, para a aquisição de 32 canais a 140 Hz por canal, o buffer teria capacidade de armazenar até 58 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 *VADSD\_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 58 segundos sem retirar dados do buffer de aquisição. Depois deste tempo ocorre *overflow* no buffer.

A função retorna no parâmetro de saída *ErrorCode* o código de ocorrência de erro durante a aquisição. Os códigos de erros são listados na tabela seguinte:

ErrorCode	Descrição
0	nenhum erro
1	overrun na interrupção
2	erro no conversor A/D
3	overflow no buffer de aquisição
4	IRQ inválida ou não disponível
100	erro de acesso ao driver
101	aquisição não está ativa
102	dimensão inválida do array do LabVIEW (ArrayLabVIEW). Número de colunas é diferente do número de canais habilitados para aquisição.

A aquisição por interrupção é finalizada quando ocorrer os erros 1 a 4.

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 *VADSD\_GetSamples* é transferido no máximo um total de 16K amostras.

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

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