

DRIVER LYNX-IO

ACESSO A I/O NO WINDOWS NT 4.0

1. INTRODUÇÃO

No Windows NT, o acesso ao I/O do microcomputador pelo programa aplicativo é bloqueado pelo sistema operacional. Desse modo as instruções **INP** e **OUT** existentes no Basic foram eliminadas no Visual Basic para Windows 95/NT. A instrução **Port** existente no Borland Delphi também foi eliminada nas versões posteriores do Delphi.

O driver LynxIO é um kernel mode driver para o Windows NT que implementa funções para o acesso ao I/O de placas para barramento ISA. Basicamente o driver LynxIO e a respectiva biblioteca DLL implementam funções para leitura e escrita de I/O.

O Lynx-IO pode ser configurado para acessar endereços de I/O a partir de um endereço base configurável pelo usuário, podendo ser utilizado para acessar I/O de placas ISA como por exemplo: ESD3201 e ESD6401.

1.1. Escopo

O objetivo deste documento é apresentar a interface com o driver realizada por módulo em Borland Delphi e Visual Basic 5.0.

1.2. Aplicabilidade

O driver LYNX-IO.SYS deve ser utilizado somente no sistema operacional Windows NT 4.0 ou compatível e com placas de expansão para barramento ISA.

	Elaboração	Aprovação
Área	SW.P&D	SW.P&D
Nome	Lauro	Lauro
Data	01/mar/1999	02/mar/1999
Visto		

Reprodução Proibida

2. INSTALANDO O DRIVER LYNX-IO

Os arquivos fornecidos para utilização do driver Lynx-IO são:

Lynx-IO.SYS	kernel mode driver para Windows NT 4.0
InstDrv.exe	programa de instalação do driver no registry do Windows NT 4.0
RegIni.exe	programa de registro dos parâmetros do driver no registry do Windows NT 4.0
Lynx-IO.ini	arquivo de registro do driver
LynxIO.DLL	biblioteca DLL de acesso ao driver
LynxAPI.pas	módulo em pascal para a interface com o driver
TESTE.EXE	programa de teste do driver
TESTE.*	arquivo de projeto do programa exemplo em Delphi 2.0
Main.*	formulário e módulo principal do programa exemplo
TesteDIO.*	formulário e módulo de teste de E/S digital
TESTE.VBP	arquivo de projeto do programa exemplo em Visual Basic 5.0
frmMain.*	formulário e módulo principal do programa exemplo em Visual Basic
LynxAPI.bas	módulo do programa exemplo em Visual Basic com a interface com a DLL

2.1. Instalando o Driver

Siga as seguintes etapas para instalar o driver LynxIO:

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:\LYNX-IO.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 LYNX-IO.INI
```

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

```
INSTDRV Lynx-IO C:\WINNT\SYSTEM32\DRIVERS\LYNX-IO.SYS
```

2.2. Como Alterar a Configuração Padrão do Driver

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

- ◆ Endereço base de I/O da placa para barramento ISA
- ◆ Número de ports de I/O da placa

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*, seleciona a janela **HKEY_LOCAL_MACHINE** e abra a pasta

SYSTEM\CurrentControlSet\Services\Lynx-IO\Parameters . Essa 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 para barramento ISA	0x340
nIoPorts	Número de ports de I/O da placa	8

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 *Lynx-IO* e clique sobre o botão *Finalizar* e em seguida sobre o botão *Iniciar*, para reiniciar o driver com os novos parâmetros.

3. INTERFACE DO DRIVER COM O PROGRAMA APLICATIVO

O acesso do programa aplicativo ao driver é disponibilizado através do módulo em Pascal LynxAPI. Esse módulo deve ser incorporado ao projeto do programa aplicativo em Borland Delphi. Se você estiver utilizando outra linguagem de programação, basta criar um módulo equivalente ao LynxAPI.pas com as referências à biblioteca LynxIO.DLL. A biblioteca DLL deverá ser copiada para o mesmo diretório do programa aplicativo.

É fornecido também um exemplo de programa aplicativo implementado em Visual Basic 5.0. As declarações necessárias para a utilização das funções da biblioteca LynxIO.DLL são fornecidas no arquivo LynxAPI.bas. Esse arquivo deve ser incluído no projeto do programa aplicativo em VB5. Para aplicações em Visual Basic, recomenda-se copiar o arquivo LynxIO.DLL no diretório do Windows NT.

4. DESCRIÇÃO DAS PRIMITIVAS DO LynxIO

Neste tópico são descritas as primitivas de acesso ao driver LynxIO. A tabela abaixo lista as primitivas da DLL:

Primitiva	Descrição
LynxIO_Open	Inicia o acesso ao driver
LynxIO_Close	Finaliza o acesso ao driver
LynxIO_DriverInfo	Informação a versão e revisão do driver
LynxIO_ReadDI_Byte	Leitura de um port de entrada da placa ISA
LynxIO_ReadDI	Leitura de dois ports de entrada da placa ISA
LynxIO_ReadDI_Byte	Escrita de um port de saída da placa ISA
LynxIO_WriteDO	Escrita de dois ports de saída da placa ISA

As funções da biblioteca DLL são descritas com a sintaxe do Object Pascal e do Visual Basic.

4.1. Primitiva LynxIO_Open

Object Pascal:

```
Function LynxIO_Open: longint;
```

Visual Basic:

```
Declare Function LynxIO_Open Lib "LynxIO.DLL" () As Long
```

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. Normalmente o erro ocorre quando o driver não foi instalado ou houve erros na parametrização do driver.

Veja também a primitiva *LynxIO_Close*.

4.2. Primitiva Lynx_Close

Object Pascal:

```
Procedure LynxIO_Close;
```

Visual Basic:

```
Declare Sub LynxIO_Close Lib "LynxIO.DLL" ()
```

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 *LynxIO_Open* deve haver uma correspondente chamada da *LynxIO_Close*.

Veja também a primitiva *LynxIO_Open*.

4.3. Primitiva *LynxIO_DriverInfo*

Object Pascal:

```
Function LynxIO_DriverInfo (Var DriverInfo: TpDriverInfo): boolean;
```

Visual Basic:

```
Declare Function LynxIO_DriverInfo Lib "LynxIO.DLL" (ByRef DriverInfo As TpDriverInfo) As Byte
```

Esta primitiva devolve no parâmetro *DriverInfo* o número da versão do driver, o endereço base de I/O da placa ISA e o número de ports de I/O. A primitiva retorna *true* se a execução foi realizada com sucesso ou *false* (zero) caso ocorra algum erro na sua execução. A estrutura *TpDriverInfo* possui a seguinte declaração em Object Pascal.

```
Type {----- Parâmetro de saída da LynxIO_DriverInfo -----}
  TpDriverInfo = record
    VersionHigh: byte;      // Versão do driver: high part
    VersionLow  : byte;      // Versão do driver: low part
    IoBase      : longint;   // Endereço base da placa
    nIoPorts   : longint;   // Número de ports de I/O
  end;
```

4.4. Primitiva *LynxIO_ReadDI_Byte*

Object Pascal:

```
Function LynxIO_ReadDI_Byte (RegNumber: byte): byte;
```

Visual Basic:

```
Declare Function LynxIO_ReadDI_Byte Lib "LynxIO.DLL" (ByVal RegNumber As Byte) As Byte
```

Esta primitiva permite a leitura de um port de entrada da placa ISA. O port a ser lido deve ser passado no parâmetro *RegNumber*. A numeração dos ports é relativa ao endereço base da placa. Por exemplo, para ler o port 341h de uma placa de endereço de endereço base 340h, deve-se passar o valor 1 no parâmetro *RegNumber*.

4.5. Primitiva *LynxIO_ReadDI*

Object Pascal:

```
Function LynxIO_ReadDI (RegNumber: byte): word;
```

Visual Basic:

```
Declare Function LynxIO_ReadDI Lib "LynxIO.DLL" (ByVal RegNumber As Byte) As Integer
```

Esta primitiva permite a leitura de dois ports de entrada da placa ISA com endereços consecutivos. O endereço do primeiro port a ser lido deve ser passado no parâmetro *RegNumber*. A numeração dos ports é relativa ao endereço base da placa. Por exemplo, para ler os ports 340h e 341h de uma placa de endereço de endereço base 340h, deve-se passar o valor 0 no parâmetro *RegNumber*. O valor lido dos ports 340h é 341h

são retornados respectivamente no byte menos significativo e no byte mais significativo do valor retornado pela função.

4.6. Primitiva LynxIO_WriteDO_Byte

Object Pascal:

```
Procedure LynxIO_WriteDO_Byte (RegNumber: byte; Value: byte);
```

Visual Basic:

```
Declare Sub LynxIO_WriteDO_Byte Lib "LynxIO.DLL" (ByVal RegNumber As Byte, ByVal Value As Byte)
```

Esta primitiva permite a escrita em port de saída da placa ISA. O dado a ser escrito no port deve ser passado no parâmetro *Value* e o número do port no parâmetro *RegNumber*. A numeração dos ports é relativa ao endereço base da placa. Por exemplo, para escrever no port 341h de uma placa de endereço de endereço base 340h, deve-se passar o valor 1 no parâmetro *RegNumber*.

4.7. Primitiva LynxIO_WriteDO

Object Pascal:

```
Procedure LynxIO_WriteDO (RegNumber: byte; Value: word);
```

Visual Basic:

```
Declare Sub LynxIO_WriteDO Lib "LynxIO.DLL" (ByVal RegNumber As Byte, ByVal Value As Integer)
```

Esta primitiva permite a escrita em dois ports de saída da placa ISA com endereços consecutivos. O dado a ser escrito nos ports deve ser passado no parâmetro *Value*. O endereço do primeiro port a ser atualizado deve ser passado no parâmetro *RegNumber*. A numeração dos ports é relativa ao endereço base da placa. Por exemplo, para escrever nos ports 340h e 341h de uma placa de endereço de endereço base 340h, deve-se passar o valor 0 no parâmetro *RegNumber*. O byte menos significativo de *Value* é escrito no primeiro port (340h) e o byte mais significativo no segundo port (341h).