TDLPortIO version 1.2 User Manual
DriverLINX Port IO Driver wrapper

The latest version is always available from the Internet WWW: `http://diskdude.cjb.net/`
If you didn't download this package from the above URL, chances are it is out-of-date. Check often!

IMPORTANT!

You should carefully read the license terms and conditions in *Section 1.0* before using this software. Your use of this software indicates your acceptance of the license agreement and warranty.

Table of Contents

1.0 License Agreement

You should carefully read the following terms and conditions before using this software. Your use of this software indicates your acceptance of this license agreement and warranty.

This license agreement does not cover the free *DriverLINX* driver provided by Scientific Software Tools (SST) Inc. All files distributed in this package, excepting those under the *\DriverLINX* tree, are covered by this agreement. Please refer to the SST agreement for the *DriverLINX* driver in the file *\DriverLINX\License.txt*

1.1 FreeWare

You may use a FreeWare Copy of this Software for life. If you find it useful, an email to the author telling him what you used it for would be nice, but not necessary. Your email address will not be given to any other party, nor will it be sold or used for profit or commercial purposes.

See *Section 6.0* for information on how to contact the author.

1.2 Distribution

You are hereby granted the right

- to make as many copies of this software and its documentation as you wish;
- give exact copies (including all files) of the original version to anyone;
- distribute the software and documentation in its unmodified form via electronic means;
- and distribute run-time files (*TDLPortIO.dll, DLPortIOX.ocx, DLPortIO.bpl, DLPortIO.dpl*) as part of your application without any royalties.

There is no charge for any of the above.

You are specifically prohibited from charging, or requesting donations, for any such copies (a small handling fee is acceptable).

You are further prohibited from distributing the software and/or documentation with other products (commercial or otherwise) without prior written permission, excluding the run-time files as described above. This is so the author can keep track of where this package is distributed.

1.3 Legal Disclaimer

THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL THE AUTHOR OR ANY OTHER PARTY WHO MAY HAVE DISTRIBUTED THE SOFTWARE AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE SOFTWARE TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

2.0 Introduction

2.1 What is the DriverLINX Port IO Driver?

The *DriverLINX* Port IO Driver allows you to use general port IO under Windows 95/98/NT. It does this by using *in* and *out* assembler instructions under Windows 95/98 (a guess, since there is no VxD), and through a kernel mode driver in Windows NT. It allows port read/writes of bytes, 16 and 32 bit words, as well as block read/writes of all ports from 100h to FFFFh.

***The DriverLINX* driver is provided freely, without support, by *Scientific Software Tools* (SST), Inc. The author of *TDLPortIO* has no relation to SST, nor did he write the *DriverLINX* driver. You can download the original *DriverLINX* driver package from the URLs below.**

Included in this package is exactly what is distributed by SST, excluding the MSVC and Visual BASIC demo code/examples which push the file to 1.5Mbytes (along with InstallShield), compared to the 57kbytes for the DLL, kernel driver (.SYS) and API information included with this package in the directory tree *\DriverLINX*.

Strictly speaking, it is not allowed to distribute the *DriverLINX* driver package other than as a whole, but the original package is too big. I have nothing to gain by *not* providing the original package and have provided ample opportunity for others to download it, from the URLs below. SST probably require that their package be distributed as a whole so their work is recognised, perhaps also why they request that their copyright messages are not to be removed. I believe that their work has been acknowledged sufficiently in this package for there not to be a problem. No copyright information has been removed from the *DriverLINX* driver files.

DriverLINX URL: `http://www.sstnet.com/ftp/unsupported/port95nt.exe`
`ftp://ftp.sstnet.com/pub/unsupported/port95nt.exe`

My Backup URL: `http://diskdude.cjb.net/files/cbuilder/DLPortIO/port95nt.exe`

2.2 What does TDLPortIO do?

TDLPortIO is an advanced wrapper. It provides a nice interface to the *DriverLINX* Port IO Driver. It will automatically handle the installation and operation of the *DriverLINX* Windows NT kernel mode driver (Windows NT accounts with administrator privileges only – see *Section 3.1*), and provides an intuitive interface to your program, like using the *Port[]* array for reading and writing ports (except ActiveX and DLL versions which require the use of functions; not indexed arrays).

2.3 What is TDLPortIO compatible with?

An attempt was made to make *TDLPortIO* compatible with the shareware Win95/NT Port IO package *TVicPort*. In addition to this, some extensions and improvements were made on their design. New features include:

- Burst read/write of a PortRec like array to include a *mixture* of Word and DWords
- Block read/write of Word and DWord buffers

TDLPortIO is limited to what the *DriverLINX* driver provides, and hence cannot be 100% compatible with *TVicPort*, as in the following example:

Hard/Soft access is not available, although the *hardaccess* property is present for code written for *TVicPort* – it simply have no effect. I assume *DriverLINX* uses the "hard access" by default, which cannot be changed. That is, it does not ask the operating system politely before accessing a port, to avoid collisions with some other driver/program. This is generally not a problem though, and would execute faster since it isn't always requesting permission from the operating system.

The only other way *TDLPortIO* does not function the same as *TVicPort* is in the *LPTBasePort* and *LPTNumPorts* properties. See *Section 4.1.4* for more information.

3.0 Installing TDLPortIO

3.1 Port IO access under Windows

Windows 95/98 allows direct port access, meaning you can use inline assembly in your code to read and write ports, much like DOS programmers are used to. Windows NT promotes itself as a more secure operating system, and as such, generally disallows the *in* and *out* CPU instructions which give access to ports. Only kernel mode processes have the privilege of using these instructions, such as the *DriverLINX* kernel mode driver included in this package.

Using the Windows NT *Service Control Manager (SCM)*, this driver is able to be *installed* into the system. Once *installed*, the driver needs to be *started* to be effective. To *remove* the driver, it needs to be *stopped* first. All four of these processes are done through the *SCM*. Only accounts with administrator privileges are able to do this however. This is why *TDLPortIO* is able to transparently install, *start*, *stop* and *remove* the *DriverLINX* driver within the *OpenDriver* and *CloseDriver* methods, **only under an administrator account**. This is how the shareware package *TVicPort* works; it will not work under a non-administrator account, according to its documentation.

The only way the *DriverLINX* kernel mode driver can be used under a normal user account is for the driver to be *installed* and *started* during bootup of the PC. Although it appears from the *Win32 Programmer's Reference* that if the driver already *installed* it can be *started* and *stopped* by a non-administrator account, it didn't work when I tried it. Thus if you intend to use the driver on a normal user account, make sure it is *installed* and configured to start "automatically" at bootup. This is the mode configured when running the installation program supplied (see Section 3.2 below), and also how *SST* configures it using their *port95nt.exe* installer outlined above.

There appears to be no conflicts if the *DriverLINX* driver is running all the time, compared to starting and stopping it as needed, as possible in an administrator account.

3.2 Automatically Installing the DriverLINX driver

You only really need to install the *DriverLINX* driver under Windows NT, since it is a kernel mode driver and needs to be registered by the system. Under Windows 95/98, you simply need the *dlportio.dll* file somewhere in your path, or in the directory where your program lies.

If you are always going to run *TDLPortIO* under an administrator account in Windows NT, you can ignore this section – when you call *OpenDriver* in the component, it will install the *DriverLINX* driver automatically (and remove it in *CloseDriver*). However, if you are going to use *TDLPortIO* under a user account *without* administrator privileges, you need to install the driver before it can be used by such accounts. *This can only be done in an administrator account due to Windows NT security.*

To automate this process, a small installation program is provided. It was designed to be as quick as possible – for people who have to do multiple installations on a variety of PCs, on many network PCs perhaps. It also fits on a 1.44 Mbyte disk, for easy transportation from PC to PC.

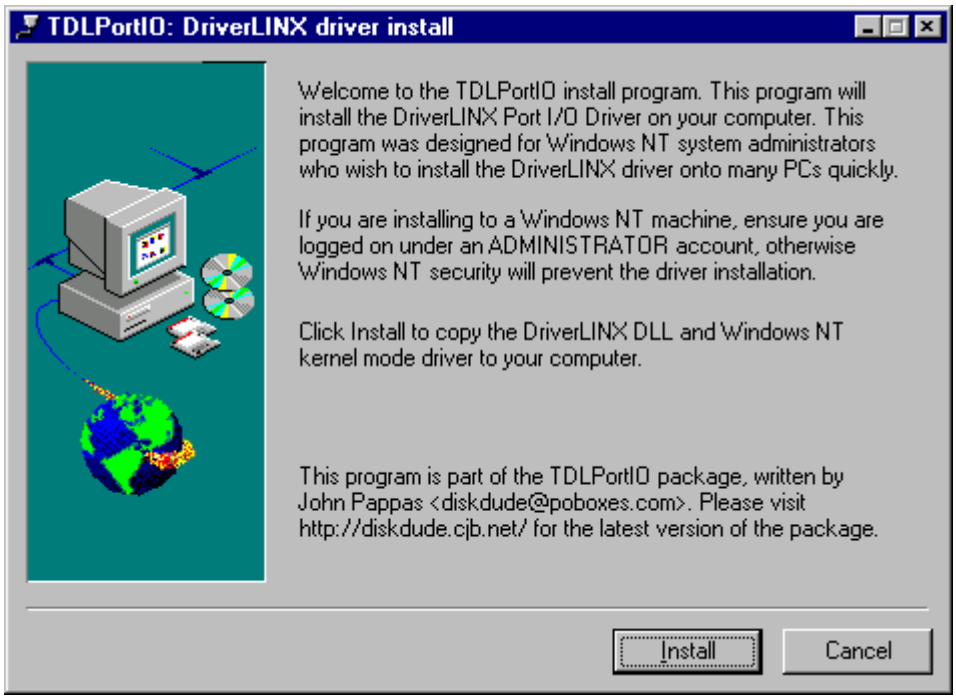

Figure 3.2 – Automatic driver installation program

Figure 3.2 shows the main screen in the automatic driver installation program. It is located in the *\install* directory of the package. Simply run the program, making sure *dlportio.dll* and *dlportio.sys* are in the same directory as the *install.exe* program. You need to copy these files from the *\DriverLINX\drivers* directory of this package.

After clicking the "Install" button, it will copy the *dlportio.dll* file into your Windows system directory. If you're running Windows NT, it will also copy the *dlportio.sys* file into your Windows drivers directory, then install the driver into the system *Service Control Manager*.

See the source in the directory *\install\source* for example C code to install the driver in your applications.

3.3 Manually Installing the DriverLINX driver

\DriverLINX\drivers\DLPortIO.dll

This is the *DriverLINX* DLL which provides Port I/O in Win95/98, and an interface to the WinNT kernel mode driver in Windows NT. Copy it to your Windows system directory.

e.g. C:\WINDOWS\SYSTEM (for Win95)
C:\WINNT\SYSTEM32 (for WinNT)

You aren't **required** to place the DLL here, but it's a good idea to do so. It will work so long as it appears in the same directory as the program using it, or somewhere in your path.

To use the DLL in some directory not in your path, use the DLLPath property. The default is a null string, meaning it will search the program's path, then the windows directory and your computer's path.

\DriverLINX\drivers\DLPortIO.sys

This is the *DriverLINX* kernel mode driver which provides Port I/O in WinNT, and as such is **only** required if you are running Windows NT.

You would normally place it in your Windows drivers directory, however the *TDLPortIO* component can locate it elsewhere through the DriverPath property, but only if running it under an administrator account (i.e. only if *TDLPortIO* will be *installing*, *starting*, *stopping* and *removing* the driver during execution).

e.g. C:\WINNT\SYSTEM32\DRIVERS

If you are only going to run *TDLPortIO* under accounts in Windows NT which don't have administrator access, you must install the driver into the system in addition to copying it to the directory outlined above. See the directory *\install\source* for some example C code of installing the driver into the Windows NT *Service Control Manager*.

Under administrator accounts, this is done automatically upon calling *OpenDriver* in the component. Removal of the driver, if installed by *OpenDriver*, occurs when calling *CloseDriver* in administrator accounts. This was done to simulate the behaviour of *TVicPort*; although *TVicPort* (shareware version) only works under administrator accounts.

Note that *TDLPortIO* will not remove a driver in *CloseDriver* that was previously installed before *TDLPortIO* was started; nor will it stop a driver which was running previously. In other words, when you call *CloseDriver*, it will return the status of the driver to what it was before *OpenDriver* was called. *CloseDriver* is automatically called when the component is destroyed, if *OpenDriver* was called previously.

3.4 Installing the C++ Builder Component

The Borland C++ Builder 3.0 component version of *TDLPortIO* is located in *\cbuilder\cbuilder.3*.
The Borland C++ Builder 4.0 component version of *TDLPortIO* is located in *\cbuilder\cbuilder.4*.

Run BCB 3.0/4.0 then select:

i. Component|Install Packages
ii. "Add" button
iii. Select the **C++ Builder 3.0/4.0** file DLPortIO.bpl (wherever you put it)
iv. "Open" button
v. "OK" button

The package is now installed and should appear in a "Diskdude" page on your component palette.

For a very small example program (mimicking the examples given in the original *DriverLINX* package) using the *TDLPortIO* component, see the *\cbuilder\cbuilder.X\demo* directory.

I will not supply a C++ Builder 1.0 version – it's time you had an upgrade! Alternatively, you have the source code, make one for yourself... :)

3.5 Installing the Delphi Component

The Borland Delphi 3.0 component version of *TDLPortIO* is located in *\delphi\delphi.3*.
The Inprise Delphi 4.0 component version of *TDLPortIO* is located in *\delphi\delphi.4*.

Run Delphi 3.0/4.0 then select:

i. Component|Install Packages...
ii. "Add" Button
iii. Select the **Delphi 3.0/4.0** file DLPortIO.bpl (wherever you put it)
iv. "Open" button
v. "OK" button

The package is now installed and should appear in a "Diskdude" page on your component palette.

For a very small example program (mimicking the examples given in the original *DriverLINX* package) using the *TDLPortIO* component, see the *\delphi\delphi.X\demo* directories.

I will not supply a Delphi 2.0 version – it's time you had an upgrade! Alternatively, you have the source code, make one for yourself... :)

3.6 Installing the ActiveX (OCX) component

You need to install *DLPortIOX.ocx* into your system with a *RegSvr32.exe* utility:

regsvr32.exe DLPortIOX.ocx

Note that in Windows ’95/'98, *regsvr32.exe* is located in your Windows System directory, which usually isn’t in your path.

3.7 Installing the ActiveX (OCX) component into Visual BASIC

Firstly you need to install the ActiveX control into your system as described above in section 3.6.

Run Visual BASIC then select:

i. Project|Components…
ii. Select “DLPrinterPortIOXControl Library” (check the item)
iii. "OK" button

The package is now installed and should appear on your component palette.

Note: When you drag the component onto your form to create it, you should set its *Visible property to false*, otherwise it will appear when you run your program.

4.0 Using TDLPortIO

4.1 As a C++ Builder, Delphi and ActiveX Component

Note that the ActiveX version only comes in one version: *TDLPrinterPortIOX*, which provides the methods and properties of *TDLPortIO* and the printer port functions within *TDLPrinterPortIO*.

4.1.1 TDLPortIO Methods

[procedure] OpenDriver()

Opens the *DriverLINX* DLL. If running under Windows NT, it will start the kernel mode driver, and if the driver isn't installed, it will attempt to install it into the system.

[procedure] CloseDriver()

Closes the *DriverLINX* DLL. If running under Windows NT, it will stop the kernel mode driver (if started in *OpenDriver*), and remove it if it was installed in *OpenDriver*.

[procedure] PortControl(TPortRec, NumberOfPorts)

Similar to the *PortControl* method provided by *TVicPort*, it will take an array of *TPortRec* structs/records, the number given by *NumPorts*, and read/write data as requested.

This is not available in the ActiveX version.

[procedure] PortCommand(TPortCommand, NumberOfPorts)

Similar to *PortControl*, this is an extension allowing reading/writing of Word and Dwords, as organised in the *TPortCommand* struct/record.

This is not available in the ActiveX version.

[procedure] ReadPortFIFO(Address, NumberOfPorts, Buffer)
[procedure] WritePortFIFO(Address, NumberOfPorts, Buffer)
[procedure] ReadWPortFIFO(Address, NumberOfPorts, Buffer)
[procedure] WriteWPortFIFO(Address, NumberOfPorts, Buffer)
[procedure] ReadLPortFIFO(Address, NumberOfPorts, Buffer)
[procedure] WriteLPortFIFO(Address, NumberOfPorts, Buffer)

Similar to the *ReadPortFIFO/WritePortFIFO* methods offered by *TVicPort*, this will read/write a block of data to a single port. The number of data items to read/write are given by *NumPorts*, and the data by the *Buffer* array. The Word and DWord versions are an extension to that provided by *TVicPort*.

4.1.2 TDLPortIO Properties

```
[Byte] Port[Address]    Read/Write
[Word] PortW[Address]   Read/Write
[DWord] PortL[Address]  Read/Write
```

Read and write these properties to access ports.

These are only available as function calls in the ActiveX version:

```
[Byte function]  ReadPort(Address)   [procedure] WritePort(Address, Data)
[Word function]  ReadPortW(Address)  [procedure] WritePortW(Address, Data)
[DWord function] ReadPortL(Address)  [procedure] WritePortL(Address, Data)
```

```
[String] DriverPath     Read/Write
```

Set this for the path to the *DriverLINX dlportio.sys* driver file when running under Windows NT in administrator mode, and the driver hasn't been installed previously. i.e. when the wrapper installs the driver, it will use this as the path to the *dlportio.sys* file.

Do not specify the filename in this property, nor any trailing \ character.

```
[String] DLLPath        Read/Write
```

Set this for the path to the *DriverLINX dlportio.dll* DLL file. If you set it to the null string, it will search the path of the program .EXE, then the Windows directory and other directories in your path.

Do not specify the filename in this property, nor any trailing \ character.

```
[Boolean] ActiveHW      Read Only
```

Read this to see if the *DriverLINX* driver is active. After a call to OpenDriver, it will read *true* if no errors occurred, otherwise it will be *false* and you can use the *LastError* property to see what happened where.

```
[Boolean] HardAccess    Read/Write
```

This property has no effect, and is here to maintain compatibility with *TVicPort*.

```
[String] LastError      Read Only
```

If an error occurred in *OpenDriver* or *CloseDriver*, and the *ActiveHW* property did not change state, this string will indicate the error which occurred.

4.1.3 TDLPrinterPortIO Methods

TDLPrinterPortIO inherits all the methods of *TDLPortIO*, and adds the following. All operations which do not specify a printer port use the "currently selected" port, selectable by the *LPTNumber* property.

[procedure] LPTStrobe

Sends a STROBE signal to the printer.

[procedure] LPTAutofd(BooleanFlag)

Sends an AUTOFD (auto line feed) signal to the printer.

[procedure] LPTInit

Resets the printer by sending INIT signal.

[procedure] LPTSlctIn

Sends SLCTIN signal to the printer.

[Boolean function] LPTPrintChar(CharacterToPrint)

Sends a character to the printer. Returns *true* on success. Repeat as necessary.

4.1.4 TDLPrinterPortIO Properties

TDLPrinterPortIO inherits all the properties of *TDLPortIO*, and adds the following:

[Boolean] Pin[PinNumber] Read/Write

When read, it returns the boolean state of the pin specified. When written, it will set that pin high (5V) or low (0V), depending on the boolean value given.

Note that a high voltage on the pin corresponds to a boolean *true*, whereas a low voltage corresponds to a boolean *false*. i.e. any inversions that the printer port does on signals is also done in software to counteract the effect.

The index range is only valid in the range 1-25. Invalid indexes are ignored.

This property is only available as function calls in the ActiveX version:

```
[Boolean function] GetPin(PinNumber)
[procedure]        SetPin(PinNumber, State)
```

[Boolean] LPTAckwl Read Only

Returns the ACKWL state from the printer.

[Boolean] LPTBusy Read Only

Returns BUSY state from the printer.

[Boolean] LPTPaperEnd Read Only

Returns PAPER END state from the printer.

[Boolean] LPTSlct Read Only

Returns SLCT state from the printer.

[Boolean] LPTError Read Only

Returns ERROR state from the printer.

[Byte] LPTNumPorts Read Only

Returns the number of printer ports installed into the *Windows* system via the *Control Panel*. This information is obtained by reading the *Windows registry*.

[Byte] LPTNumber Read/Write

Selects the printer port for use with all the methods and properties above. Ports are identified by their number (i.e. 1, 2, 3, etc), **not** by their port address.

Note that port numbers are those assigned by *Windows* in the *Control Panel*.

[Word] LPTBasePort Read Only

Returns the base address of the current printer port being operated on.

Similarly to the *LPTNumPorts* property, this information is obtained from the *Windows registry*.

4.2 As a DLL

The DLL version implements all the methods and properties mentioned for both *TDLPortIO* and *TDLPrinterPortIO* above. Properties which are *Read/Write* are split into two functions, *GetXXX*, and *SetXXX* which return the current value (read) of the property, and write a new value of the property respectively, where *XXX* is the property name.

All functions use the standard Windows *stdcall* calling convention.

5.0 Bugs, To Do List, and Version History

5.1 Found a bug? Changed the source?

If you find a bug in the code, or make any changes, it would be appreciated if you could email me <diskdude@poboxes.com>, so that I may review the change(s), and modify the official distribution and make it available to others.

Make sure you have downloaded the latest version before reporting bugs. The bug may have already been fixed!

5.2 To Do List

The *TDLPortIO* package has had many improvements since the first version in January 1999. It does everything I originally wanted it to do, and much much more. I only wanted a simple wrapper for C++ Builder for a small project I was working on. I had never imagined to write the Delphi, ActiveX and DLL versions too! It got to a point where I stopped working on my other project, and started working on *TDLPortIO* alone…

It is envisaged that no further improvements will be made to the *TDLPortIO* package, except perhaps fixes to any bugs that people may discover.

5.2 Version History

v1.2 – 10th July 1999

- Thanks to *Peter Holm* <comtext3@post4.tele.dk>, detection of the number of printer ports and their base addresses is now possible in the *TDLPrinterPort* component, and DLL.

  He managed to hack the *Windows Registry*, and work out where the relevant information is stored, for both Windows '95/98 and Windows NT. The detection code in this package is based on his algorithm and code. Note that I used the Windows API functions for all access to the registry, since the registry wrapper object in Delphi 3 and C++ Builder 3 does not allow read only access to keys; the Windows NT code does not work without it.

  A few small modifications were also made to Peter's original algorithm/code so it worked properly on all tested systems.

v1.13 – 3rd April 1999

- I cannot afford a PO BOX any more; *TDLPortIO* is now FreeWare!

- All indexed properties in the ActiveX control, when used, would result in Access Violations. To avoid this, they have been converted into methods. I don't know how to create ActiveX control indexed properties using C++ Builder. If someone can help, it would be appreciated.

- The C++ code now looks a little cleaner, and faster as some small functions in the *TDLPrinterPortIO* component have been put inline.

- Fixed a bug in the installation program where it would report an error when the *DriverLINX* driver was already installed into the Windows NT *Service Control Manager*.

v1.12 – 22nd January 1999

- Fixed up the documentation and license agreement to make it more clear that I did not write the *DriverLINX* driver by SST, and thus it does not come under my license agreement.

  All files included in this package *other than those in the \DriverLINX tree* come under the agreement in *Section 1.0*. I hate all this legal stuff, and next time I think I'll stick to a GNU copyleft agreement… with postcards highly recommended, but not mandatory.

v1.11 - 17th January 1999

- Changed the access permissions requested when connecting to the Windows NT *Service Control Manager*, so users on accounts without administrator privileges can open the *DriverLINX* driver if it is already installed on the system.

- Added a program to install the *DriverLINX* driver on a Windows NT system (so users without administrator privileges can use *TDLPortIO*). It will make the driver active at bootup, the only way non-administrator accounts can use the driver. Do not use this program if you will only run *TDLPortIO* under an account with administrator privileges.

- Added to, and converted, the documentation to this PDF file.

v1.10 - 10th January 1999

- Fixed a bug in the *Pin*[] property. It always set the pin true, never what it was told to set.
- Fixed a bug in the *LPTBasePort* property where it always returned 378H
- Fixed a bug in *LPTAutofd*(*bool*) where it would always set the Auto LF state of the printer to *false*, not to what it was told to set it to.
- Added a bit more/better error checking/correction in *OpenDriver*()
- All inverted pins on the printer port are now inverted in software also, for the *Pin*[] property only. This means if you set any of the pins to true, and it is an output pin, it means the pin will be at 5V. Setting a pin to false will lower the pin to 0V.

  This was done to further maintain compatibility with *TVicPort*.
- Disabled Pin[] writes on pins 10,11,12,13,15 since these are the control line inputs, and to again maintain compatibility with TVicPort.
- Added a native Delphi version
- Added an ActiveX (OCX) version
- Added a DLL version

v1.01 - 8th January 1999

- Renamed the *TVicHW32PortRec* type to *TPortRec*
- Split the *TDLPortIO* class into two classes/components:
  a) *TDLPortIO*
  b) *TDLPrinterPortIO* (derived from *TDLPortIO*)

  This is more like how the components are arranged in the *TVicPort* package. Again, to make it more compatible.

  I was going to modify the definition of the *TPortRec* to the Delphi-compatible one from port_32.hpp from the *TVicPort* package (which was converted by C++ Builder from a pascal version, and is used in the C++ Builder version of *TVicPort*), but I feel that the definition I've given is more friendly to C programmers, and uses less memory.
- You no longer have to have the *DriverLINX* driver installed into Windows NT before you can use the wrapper. It is installed and started automatically when you call *OpenDriver*(), and stopped and removed automatically when you call *CloseDriver*() – or if you don't call it directly, it will be called for you in the destructor.

  Under Windows 95/98, as before, the DLL isn't loaded until you call *OpenDriver*() and is unloaded when you call *CloseDriver*().

- No longer is *DLPortIO.sys* required to be in *<windows system>\DRIVERS* under Windows NT. The new property 'AnsiString DriverPath' will let you specify the path (NOT the filename of the .SYS). This means you can have both the .DLL and .SYS sitting in the same directory, such as the same directory as the main .EXE for example.

v1.00 - 6th January 1999

- Initial version

6.0 Contacting the Author

This used to be the section where the CardWare concept was described, with the address of my PO BOX. I can no longer afford one, so *TDLPortIO* is now FreeWare (not public domain; I retain the copyright), instead of CardWare.

I used to collect PostCards from around the world. You can see them (which were scanned in) on the WWW: `http://diskdude.cjb.net/postcards/`

If you think *TDLPortIO* is really, really good, why not email me and let me know? I like hearing about how people use my software - it encourages me to write more!(:

My Internet email address is: `diskdude@poboxes.com`

Note that I am not getting anything for this software. Any support which I may give via email is at my expense, during my spare time. You may not get a reply if you ask a stupid question, or if you do not read this entire manual first.

7.0 Thanks!

A big thanks to Peter `<petert@minerva.com.au>` for his testing of the *DriverLINX* driver and this wrapper in his copy of Windows NT. Since I don't run NT myself, I would have not been able to debug the automatic driver loading/unloading code without his help, and prototype board! (:

Another big thanks to Peter Holm `<comtext3@post4.tele.dk>` for his printer port detection algorithm and code.

8.0 Trademarks