

## How to communicate with LPT1 on an NT Operating System

The NT operating system prevents direct hardware manipulation since it violates the general protection scheme. There are several pieces of code available to allow access to IO ports that I will discuss below.

The first method is an older piece of code that gives IO permission to the calling module at a base level. Once registered the system IO can be handled in the usual way from programming code as if the system is running WIN9x OS. The software is listed below and is called GIVEIO. This code is freeware.

<http://users.skynet.be/k-net/ParPort/>

The second method found on the WEB is from Scientific Software Inc. and is called DriverLINX. This code uses a DLL library to make calls to and from the IO device. This method allows a more convenient way to make calls from within applications such as Excel. Examples of using this driver inside Excel will be shown. Software can be downloaded from <http://www.sstnet.com/> The program name is (port95nt.exe)

The DLL allows byte, word and long word reads and writes to IO ports without protection violations. After placing the DLL dlportio.dll in the system area calls can be made as seen in the listing below.

```
Public Declare Function DlPortReadPortUchar Lib "dlportio.dll" (ByVal Port As Long) As Byte
Public Declare Sub DlPortWritePortUchar Lib "dlportio.dll" (ByVal Port As Long, ByVal Value As Byte)

Sub OnRead_Click()
    Dim Value As Long
    Dim WValue As Long
    WValue = Cells(4, 2).Value
    DlPortWritePortUchar 888, WValue
    Value = DlPortReadPortUchar(888)
    textvalue = "&H" + Hex(Value)
    Cells(2, 2).Value = textvalue
    Cells(2, 3).Value = Value
End Sub
```

## How to communicate with LPT1 on an NT Operating System

### Additional information on programming the parallel port

#### Programming the parallel port under Windows NT/2000

Porting their applications from 16 to 32 bits for use with win95 and NT, many people are confronted to the problem of accessing the ports. Windows NT indeed prevents an application to access the hardware, and the common functions for IO access used in 16 bits sources are not present any more when you want to compile 32 bits (at least this is the case with BC++5).

WARNING: This page is rather old and possibly not up-to-date. You can alternatively check [Jan Axelson's pages about the parallel port](#). I've not completely solved the problem now, but here are pieces of solution:

◆ Writing a driver. This is the best solution but it's not accessible to the novice I am.

◆ TinyPort is a 70\$ shareware that gives access to a restricted number of ports. It comes with a DLL that interfaces your program with the driver. I have not tested it myself but here is what it's author says: "And perhaps you might consider mentioning the big difference between TinyPort and the other drivers you list: TinyPort is a full-blown kernel mode driver that follows the guidelines MS has set up, so it won't undermine the stability of NT". [Download TinyPort](#)

◆ GiveIO is a rather old freeware. But sometimes old stuff is the best... This small driver removes the NT protection by giving Hardware access rights to a process that calls it. See tstio.c in the archive. [Download](#) it from here. You can install the driver very easily using Microsoft DDK or a small program that will do everything for you: loaddrv, also available for [download](#). Before running your application, you should stop the parallel port service by typing NET STOP PARPORT from a command line window. You can then access the ports as you would in Win95.

◆ VicPort. An *excellent* shareware driver that comes with examples, a DLL, and components for Delphi and C++ Builder !!! The light version allows one to access the ports, the full version lets you access the ports, the memory and the IRQ... What do you want more?

[Download](#) full version

[Download](#) light version ( works very well for parallel port or other port operations )

---

Samuel Grimee  
[sam@linuxbe.org](mailto:sam@linuxbe.org)