

UserPort Documentation

1 History

UserPort V1.0:

First release.

UserPort V1.1:

Changes from version 1.0 to 1.1:

- Added Test LPT1 button
- Added Test Spkr button
- Start button is grayed when the driver is started
- Stop button is grayed when the driver is stopped
- UserPort driver now works on Multiprocessor machines and Hyperthreading machines
- UserPort now works on all Windows XP machines
- Settings are now stored in HKLM\System\CurrentControlSet\Services\UserPort because the driver failed to read the key HKLM\Software\UserPort
- \\.\UserPort access is now stopped when the \\.\UserPort file is closed
- Sleep(100) is no longer needed after the CreateFile("\\.\UserPort"...) call
- Address range is extended to 0x7a80 for all processes and to 0xffff for \\.\UserPort processes
- Added Java and VisualBasic example files
- Removed HandlerExceptionFilter from C example file
- Updated C and PAS example files
- The registry settings could be corrupted on certain circumstances, this bug has been solved
- UserPort did not run stable on Hyperthreading machined (rebooted without bluescreen). Solved by using the hard coded value of 0x20AC for the IOPMPtr variable on Windows XP/2000 machines

UserPort V2.0:

Changes from version 1.1 to 2.0:

- Changed to PsSetCreateProcessNotifyRoutine method
- Removed the ugly and assembler modification of the TSS (not needed now with the process notification routine)
- Changed to only one IOPM registry map
- Address range is extended to 0xffff for all processes

2 Summary

UserPort.SYS is a kernel mode driver for Windows NT/2000/XP that gives user mode programs access to I/O Ports. This makes it possible to access hardware directly from a normal executable in the same way as under Windows 95/98/ME. This driver does not work on Windows 95/98/ME and there is really no need to run it anyway because I/O ports are always granted to user mode programs on these operating systems.

The driver can be used for the following purposes:

- To run software on Windows NT/2000/XP that normally only runs on Windows 95/98/ME.
- To easily access hardware like the parallel port and other I/O ports.

So what's the drawback with this wonderful software? Microsoft has for security reasons prohibited user mode access to I/O ports. Opening up I/O ports creates a big security hole in your system. You should therefore carefully set the grant lists to only give user mode access to the specific I/O ports you need. The default values opens up a wide range of I/O ports and you should narrow it down.

The correct driver approach for a commercial product is to create a driver that communicates with the hardware and the communication between the driver and your application should be done through CreateFile/ReadFile/WriteFile/DeviceIoControl. But this is not my aim with the driver. My aim with the driver is listed above. There are a lot of parallel port drivers out there which gives control to user-mode applications in a nice way.

3 Installation

The driver can be installed in the following two ways:

- Copy `UserPort.SYS` to `%WINDIR%\SYSTEM32\DRIVERS`
Start `UserPort.EXE` and add the addresses you want and remove the others and click on start.
- Run `UserPort.EXE` with the driver filename and path as an argument
i.e. run `UserPort.EXE X:\YOURDIR\UserPort.SYS`
Add the addresses you want and remove the others and click on start.

You should now have usermode access to the addresses you have chosen.

5 Testing UserPort

You can verify that `UserPort` has been installed and loaded correctly by clicking on the “Test LPT1” and the “Test Spkr” buttons on `UserPort.EXE`. The “Test LPT1” button will reset your printer and the “Test Spkr” will play a short tune on your internal speaker. Note that you need to add the port 42, 43 and 61 before you try the “Test Spkr” button.

4 Examples

Port instructions are not included in development environments (such as Visual C++ and Delphi) because direct I/O access isn't allowed by the operating system. You will therefore need to include a portion of assembler code into your software in order to access your hardware, see Figure 1, 2 and 3.

```
char inportb(int portid)
{
    char value;

    __asm mov edx,portid
    __asm in al,dx
    __asm mov value,al
    return value;
}
```

Figure 1: Read I/O port

```
void outportb(int portid,
char value)
{
    __asm mov edx,portid
    __asm mov al,value
    __asm out dx,al
}
```

Figure 2: Write I/O port

```
char value;
value = inportb(0x37a);
value = value & 0xfb;
outportb(0x37a,value);
Sleep(100);
value = value | 0x04;
outportb(0x37a,value);
```

Figure 3: Reset a printer connected to LPT1

Figure 3 shows how simple it now is to access hardware from a usermode program. The `UserPort` package contains examples for developing C, C++, Delphi, Java and Visual Basic applications.

5 Technical Description

The driver gives user mode program access to selected ports by changing the x86-processors IOPM (I/O Permission Map). Figure 1 shows how the driver works. For a detailed description on the TSS see Intel processor handbooks.

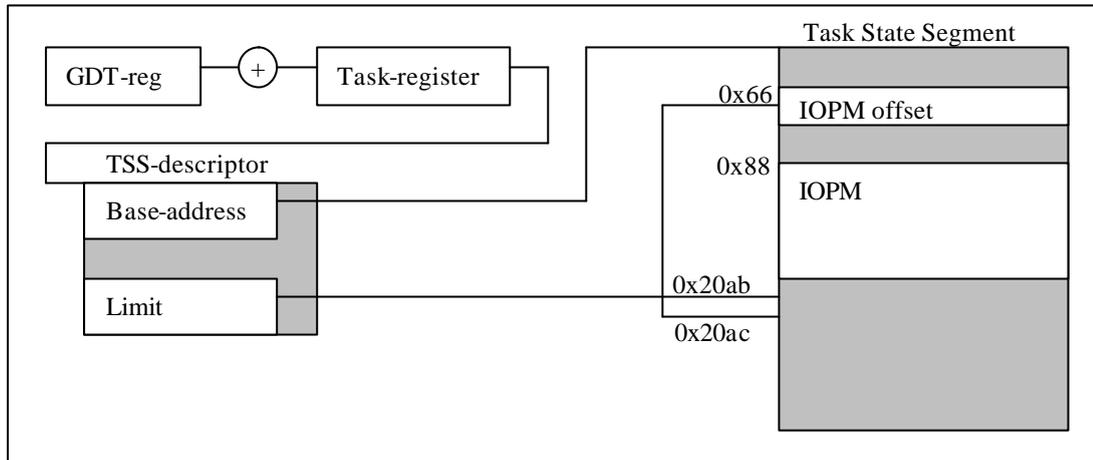


Figure 4: 80x86 TSS description

The IOPM is a memory map where every bit is used to grant or deny port access when running the CPU in user mode. Intel has reserved an IOPM area at offset 0x88 but the system can use other locations as well. This area is 0x2000 bytes which can then control the full 16 bit port address range ($0x2000 \times 8 = 65536$). A '1' means no access and a '0' means full access. The default value of the IOPM is only '1's, also the default IOPM offset is 0x20ab which is outside the total TSS size of 0x20ab which again prohibits all user mode port IO. Simply changing the IOPM offset by writing 0x88 does not work since the OS rewrites this memory location on every task switch. But there are undocumented kernel functions for this purpose! Change the IOPM offset to 0x88 is done by a simple call to the `Ke386IoSetAccessProcess` function. The `Ke386IoSetAccessProcess` function needs to be called for every process which needs direct IO access. The UserPort driver calls the `Ke386IoSetAccessProcess` function every time a new process is created by registering a process notify function through the `PsSetCreateProcessNotifyRoutine` kernel function call.

The user mode part of UserPort (UserPort.EXE) takes care of setting up the allowed ports and installs the kernel mode driver (UserPort.SYS). The user configurable IOPM map is stored in the registry at:
`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\UserPort\IOPM`

This driver is influenced and inspired by an article written by Dale Roberts 8/30/95, published in May 96 Dr Dobbs Journal, see www.ddj.com.

6 License

This software is released as freeware. You may use it for personal or commercial use. You may distribute it but I would like you to keep the installation package intact. If you make modifications to the driver, please send me the modified version and I add your modifications into the next release.

If you want to only distribute the UserPort.sys driver and have your installation package installing the driver and creating the registry keys you may do so.

If you like this program, please send me a postcard from your hometown!

Tomas Franzon
Tröskaregatan 12
583 34 Linköping
Sweden
tomas_franzon@hotmail.com