

# Table of Contents

<b>1 Introduction</b> .....	1-1
<b>2 DOS / Windows 3.x Installation</b> .....	2-1
2.1 DSP-100 Client Driver for DOS .....	2-2
2.1.1 Client Driver Installation .....	2-2
2.1.2 Command Line Options .....	2-3
2.1.3 Common Problems .....	2-7
2.2 DSP-100 Enabler for DOS .....	2-8
2.2.1 Command Line Options .....	2-9
2.2.2 Common Problems .....	2-12
<b>3 Windows 95 Installation</b> .....	3-2
3.1 Installing a DSP-100 Under Windows 95. ....	3-2
3.2 DSP-100 Resource Settings in Windows 95 .....	3-3
3.2.1 Viewing Resource Settings with Device Manager .....	3-3
3.2.2 Changing Resource Settings with Device Manager .....	3-5
<b>4 OS/2 Installation</b> .....	4-1
4.1 Command Line Options .....	4-2
4.1.1 Configuring With "System Assigned" Resources .....	4-2
4.1.2 Configuring With "User Assigned" Resources .....	4-3
4.1.3 Advanced Configuration Topics .....	4-4
4.2 Monitoring The Status Of PCMCIA Cards .....	4-5
4.3 Common Problems .....	4-5
<b>5 Unix Support</b> .....	5-1
5.1 Unix Enhanced Serial Device Driver .....	5-1
<b>6 Hardware Information</b> .....	6-1
<b>7 External Connections</b> .....	7-1
<b>8 Specifications</b> .....	7-1

# List of Figures

Figure 1. Client Driver vs. Enabler for DOS/Windows 3.x. . . . .	2-1
Figure 2. Windows 95 Device Manager . . . . .	3-4
Figure 3. DSP-100 Basic Configuration Table. . . . .	3-5
Figure 4. Windows 95 Resource Allocation . . . . .	3-6
Figure 5. Block Mode Addressing. . . . .	5-1
Figure 6. Interrupt Status Register. . . . .	5-1
Figure 7. DSP-100 Adapter Cable. . . . .	6-1
Figure 8. Standard RS-232 Signal Assignment. . . . .	6-1

# 1. Introduction

---

The DSP-100 provides two independent RS-232 asynchronous serial communications interfaces for systems equipped with PCMCIA Type II and/or Type III expansion sockets. The DSP-100 is a PCMCIA Type II (5 mm) card and is PCMCIA PC Card Standard Specification 2.1 compliant.

The DSP-100's serial ports are implemented using 16C550 Universal Asynchronous Receiver/Transmitters (UARTs) which are the recommended communications interface for multitasking environments and with applications involving high data transfer rates.

The DSP-100's two serial ports are addressable in two modes:

1. **'Block Mode'**: The two serial ports are configured in one 16-byte continuous block of I/O address space. The block must begin on an even 16-byte division. Both serial ports share one IRQ level.
2. **'COM Mode'**: The two serial ports are configured at the standard COM port I/O address locations. The ports may be configured as COM1 and COM3; also, the ports may be configured as COM2 and COM4. Both serial ports share one IRQ level.

A special interrupt status register is also available to simplify the software required to service multiple serial ports in an interrupt driven environments. See the **Hardware Information** section for details.

(This page intentionally left blank.)

## 2. DOS / Windows 3.x Installation

---

Two configuration software programs are provided with the DSP-100: a Client Driver, DSP100CL.SYS, and a card Enabler, DSP100EN.EXE. Both of these programs are executed from DOS (before entering Windows) and allow operation of the DSP-100 in both the DOS and Windows 3.x environments. For optimal operation, however, the Client Driver is the preferred method of installation and configuration. The table below highlights the differences between these programs.

Client Driver (recommended)	Enabler (not recommended)
File name: DSP100CL.SYS	File name: DSP100EN.EXE
File type: DOS device driver	File type: DOS executable
Interfaces to PCMCIA Card and Socket Services software (PCMCIA host adapter independent)	Interfaces directly to Intel 82365SL and other PCIC compatible PCMCIA host adapters
Allows automatic configuration of DSP-100 adapters upon insertion (Hot Swapping)	Does not support automatic configuration of DSP-100 adapters upon insertion (Hot Swapping)
Requires PCMCIA Card and Socket Services software	Does not require PCMCIA Card and Socket Services software

Figure 1. Client Driver versus Enabler for DOS/Windows 3.x.

Card and Socket Services software is commercially available from several vendors for most desktop and laptop PCs. If you are unsure whether Card and Socket Services software is currently installed on your system, install the DSP-100 Client Driver as discussed in following section. When loaded, the Client Driver will display an error message if Card and Socket Services software is not detected.

## 2.1 DSP-100 Client Driver for DOS

In order to use the DSP-100 Client Driver, the system must be configured with Card and Socket Services software. Card and Socket Services software is not provided with the DSP-100 but is available from Omega.

### **IMPORTANT:**

Some versions of Card and Socket Services dated before 1993 do not support general purpose I/O cards. If after careful installation of the Client Driver the DSP-100 does not configure or operate properly, an updated version of Card and Socket Services may be required. Card and Socket Services software is available from Omega Inc.

### 2.1.1 Client Driver Installation

The following procedure is used to install the DSP-100 Client Driver:

1. Copy the file DSP100CL.SYS from the DSP-100 distribution diskette onto the system's hard drive.
2. Using an ASCII text editor, open the system's CONFIG.SYS file located in the root directory of the boot drive.
3. Locate the line(s) in the CONFIG.SYS file where the Card and Socket Services software is installed.
4. **AFTER** the line(s) installing the Card and Socket Services software, add the following line to the CONFIG.SYS file:

```
DEVICE = drive:\path\DSP100CL.SYS options
```

where *options* are the DSP-100 Client Driver command line options discussed on the following pages.

5. Save the CONFIG.SYS file and exit the text editor.
6. Insert the DSP-100 into one of the system's PCMCIA slots.

**NOTE:** Since the DSP-100 Client Driver supports "Hot Swapping", it is not necessary to have the DSP-100 installed when booting the system. By inserting the card before booting, however, the Client Driver will report the adapter configuration during the boot process thereby verifying the changes made to the CONFIG.SYS.

7. Reboot the system and note the message displayed when the DSP-100 Client Driver is loaded. If the Client Driver reports an "invalid command line option", correct the entry in the CONFIG.SYS file and reboot the system again. If the Client Driver reports "Card and Socket Services not found", a version of Card and Socket Services must be installed on the system or the DSP-100 Enabler program must be used to configure the adapter. If the Client Driver reports the desired adapter configuration, the installation process is complete and the DSP-100 may be removed and / or inserted from the system as desired. On each insertion into the PCMCIA socket, the DSP-100 will be automatically reconfigured according to the command line options.

### 2.1.2 Command Line Options

The DSP-100 Client Driver accepts up to eight command line arguments from the user to determine the configuration of the DSP-100. If any arguments are provided, the Client Driver will attempt to configure any DSP-100s with the options specified in the order they are entered on the command line. Each argument must be enclosed in parenthesis and must be separated from other arguments by a space on the command line. Within each argument, any or all of the following parameters may be specified using a comma (no spaces) to separate each parameter:

- Baddress* specifies a "block mode" base I/O address of the DSP-100 in hexadecimal. This address must reside on an even 16-byte (10H) boundary. This option **must** be omitted if using the *Dmode* option. If both the *Dmode* and *Baddress* options are omitted, a "block mode" base address will be assigned by Card and Socket Services.
- Dmode* specifies a "COM" mode" base I/O address configuration for the DSP-100. The **D1** option configures the DSP-100 at COM1/COM3 (3F8 / 3E8), and the **D2** option configures the DSP-100 at COM2/COM4 (2F8 / 2E8). This option **must** be omitted if using the *Baddress* option. If both the *Dmode* and *Baddress* options are omitted, a "block mode" base address will be assigned by Card and Socket Services.
- Iirq* specifies the interrupt level (IRQ) of the DSP-100 in decimal. *irq* must be one of the following values: 3, 4, 5, 7, 9, 10, 11, 12, 14, 15, or 0 if no IRQ is desired. If this option is omitted, an interrupt level will be assigned by Card and Socket Services.

*Ssocket* specifies which PCMCIA socket the DSP-100 must be inserted into for this configuration argument to be used. *socket* must be in the range 0 - 15. If this option is omitted, the configuration argument will apply to DSP-100s inserted into any socket.

U instructs the Client Driver to disable the DSP-100's interrupt status register and enable the Scratchpad registers of the individual UARTs. This option is only required in very rare cases where an application program requires access to the UART's Scratchpad register. If this option is omitted, the DSP-100's interrupt status register is enabled and the UARTs' Scratchpad registers are disabled.

#### 2.1.2.1 Example 1

```
DEVICE = C:\DSP-100\DSP100CL.SYS
```

In example 1, no command line arguments are specified. The Client Driver will configure a DSP-100 inserted into any socket in "block mode" with a base address and IRQ assigned by Card and Socket Services. The DSP-100's interrupt status register will be enabled.

#### 2.1.2.2 Example 2

```
DEVICE = C:\DSP-100\DSP100CL.SYS (d1)
```

In example 2, a single command line argument is provided. The Client Driver will attempt to configure a DSP-100 inserted into any socket in "COM mode" at COM1/COM3 and an IRQ assigned by Card and Socket Services. If address COM1 (3F8) or COM3 (3E8) are unavailable, the DSP-100 will not be configured. If the Client Driver can successfully configure the DSP-100 its interrupt status register will be enabled.



### 2.1.2.3 Example 3

```
DEVICE = C:\DSP-100\DSP100CL.SYS (s0,b300,i5)
```

In example 3, a single command line argument is provided. The Client Driver will attempt to configure a DSP-100 inserted into socket 0 with a base address of 300H and IRQ 5. If address 300H or IRQ 5 is unavailable, the DSP-100 will not be configured. In addition, if a DSP-100 is inserted into any other socket, it will not be configured. If the Client Driver can successfully configure the DSP-100 its interrupt status register will be enabled.

### 2.1.2.4 Example 4

```
DEVICE = C:\DSP-100\DSP100CL.SYS (i5,u,b300)
```

In example 4, a single command line argument is provided. Because the parameter order is not significant, the Client Driver will attempt to configure a DSP-100 inserted into any socket with a base address of 300H and IRQ 5. If address 300H or IRQ 5 is unavailable, the DSP-100 will not be configured. If the Client Driver can successfully configure the DSP-100, its interrupt status register will be disabled (Scratchpad registers enabled).

### 2.1.2.5 Example 5

```
DEVICE = C:\DSP-100\DSP100CL.SYS (b300,i5) (i10) ( )
```

In example 5, three command line arguments are provided. The Client Driver will first attempt to configure a DSP-100 inserted into any socket with a base address of 300H and IRQ 5. If address 300H or IRQ 5 is unavailable, the Client Driver will proceed to the second command line argument and attempt to configure the card with a base address assigned by Card and Socket Services and IRQ 10. If IRQ 10 is also unavailable, the Client Driver will proceed to the third command line argument and attempt to configure the DSP-100 with a base address and an IRQ assigned by Card and Socket Services. If the DSP-100 is successfully configured, its interrupt status register will be enabled.

### 2.1.2.6 Example 6

```
DEVICE = C:\DSP-100\DSP100CL.SYS (b300,i5) ( ) (i10)
```

In example 6, the three command line arguments of example 5 have been rearranged. The Client Driver will first attempt to configure a DSP-100 inserted into any socket with a base address of 300H and IRQ 5. If address 300H or IRQ 5 is unavailable, the Client Driver will proceed to the second command line argument and attempt to configure the card with a base address and IRQ assigned by Card and Socket Services. Since the second command line argument includes all available address and IRQ resources, the third command line argument will never be reached by the Client Driver. It is the user's responsibility to place the command line arguments in a logical order.

### 2.1.2.7 Example 7

```
DEVICE = C:\DSP-100\DSP100CL.SYS (s0,b300,i5) (s1,b340,i10)
```

The type of configuration shown in example 7 may be desirable in systems where more than one DSP-100 is to be installed. In this example, the Client Driver will attempt to configure a DSP-100 inserted into socket 0 with a base address of 300H and IRQ 5. If the DSP-100 is inserted into socket 1, the Client Driver will attempt to configure it with base address 340H and IRQ 10. This allows the user to force the DSP-100's address and IRQ settings to be socket specific which may simplify cable connections and software development. As in the previous examples, however, if the requested address or interrupt resources are not available, the DSP-100 will not be configured.

### 2.1.3 Common Problems

#### **Generic Client Drivers:**

Many Card and Socket Services packages include a generic client driver (or SuperClient) which configures standard I/O devices. If one of these generic client drivers is installed, it may configure the DSP-100 causing the DSP-100 client driver to fail installation. In these cases, the user should do one of the following:

1. modify the operation of the generic client driver to disable the configuration of modem/serial port cards. Consult the Card and Socket Services documentation for availability and details of this feature.
2. place the DSP-100 client driver before the generic client driver in the CONFIG.SYS.

#### **Available Resources:**

One function of the Card and Socket Services software is to track which system resources (memory addresses, I/O addresses, IRQs, etc.) are available for assignment to inserted PCMCIA cards. Sometimes, however, the Card Services software assumes or incorrectly determines that a particular resource is used when it is actually available. Most Card and Socket Services generate a resource table in a file (typically in the form of an .INI file) which the user can modify to adjust the available system resources. Consult the Card and Socket Services documentation for availability and details of this feature.

#### **Multiple Configuration Attempts:**

Some Card and Socket Services have a setting which aborts the configuration process after a single configuration failure (such as a request for an unavailable resource). The user should change this setting to allow for multiple configuration attempts. Consult the Card and Socket Services documentation for availability and details of this feature.

#### **Older Versions of Card and Socket Services:**

Some versions of Card and Socket Services dated before 1993 do not support general purpose I/O cards. If after careful installation of the Client Driver the DSP-100 does not configure or operate properly, an updated version of Card and Socket Services may be required. Card and Socket Services software is available from Omega.

## 2.2 DSP-100 Enabler for DOS

For systems that are not operating PCMCIA Card and Socket Services software, the DSP-100 DOS Enabler may be used to enable and configure the adapter. This Enabler, DSP100EN.EXE, will operate on any DOS system using an Intel 82365SL or PCIC compatible PCMCIA host adapter including the Cirrus Logic CL-PD6710 / 6720, the VLSI VL82C146, and the Vadem VG-365 among others.

### **IMPORTANT:**

In order to use the DSP-100 Enabler for DOS, the system **MUST NOT** be configured with Card and Socket Services software. If a Card and Socket Services software is installed, the DSP-100 Enabler may interfere with its operation and with the device(s) it controls.

The DSP-100 Enabler does not support automatic configuration of adapters upon insertion, more commonly referred to as "Hot Swapping". This means the adapter must be installed in one of the system's PCMCIA sockets before executing DSP100EN.EXE. If more than one adapter is installed in a system, the Enabler must be executed separately for each adapter. Furthermore, DSP100EN.EXE should be executed to release the resources used by the adapter before it is removed from the PCMCIA socket. Since PCMCIA adapters do not retain their configuration after removal, any adapter that is removed from the system must be reconfigured with the Enabler after re-inserting it into a PCMCIA socket.

### **IMPORTANT:**

The Enabler requires a region of high DOS memory when configuring a DSP-100. This region is 1000H bytes (4KB) long and by default begins at address D0000H (the default address may be changed using the "W" option). If a memory manager such as EMM386, QEMM, or 386Max is installed on the system, this region of DOS memory must be excluded from the memory manager's control. Consult the documentation provided with the memory manager software for instructions on how to exclude this memory region.

## 2.2.1 Command Line Options

To configure a DSP-100 in the system, the Enabler requires one command line argument from the user to determine the configuration of the card. This argument must be enclosed in parenthesis and within the argument, any or all of the following parameters may be specified using a comma (no spaces) to separate each parameter:

- Ssocket* specifies which PCMCIA socket the DSP-100 must be inserted into for this configuration argument to be used. *socket* must be in the range 0 - 15. This option is always required.
- Baddress* specifies a "block mode" base I/O address of the DSP-100 in hexadecimal. This address must reside on an even 16-byte (10H) boundary. Specify only **one** of the following three options: *Baddress*, *Dmode*, or 'R'. Use of one of these options is **always** required.
- Dmode* specifies a "COM" mode" base I/O address configuration for the DSP-100. The 'D1' option configures the DSP-100 at COM1/COM3 (3F8 / 3E8), and the 'D2' option configures the DSP-100 at COM2/COM4 (2F8 / 2E8). Specify only **one** of the following three options: *Baddress*, *Dmode*, or 'R'. Use of one of these options is **always** required.
- Iirq* specifies the interrupt level (IRQ) of the DSP-100 in decimal. *irq* must be one of the following values: 3, 4, 5, 7, 9, 10, 11, 12, 14, 15, or 0 if no IRQ is desired. This option is required if the 'R' option is not used.
- Waddress* specifies the base address of the memory window required to configure the DSP-100. Set *address* = D0 for a memory window at segment D000, *address* = D8 for a memory window at segment D800, etc. Valid settings for *address* are C8, CC, D0, D4, D8, and DC. If this option is omitted, a memory window at segment D000 will be used.
- U instructs the Enabler to disable the DSP-100's interrupt status register and enable the Scratchpad registers of the individual UARTs. This option is only required in very rare cases where an application program requires access to the UART's Scratchpad register. If this option is omitted, the DSP-100's interrupt status register is enabled and the UARTs' Scratchpad registers are disabled.

Before removing a DSP-100 from its PCMCIA socket, the Enabler should be executed to free the system resources allocated when the card was installed. For this operation the Enabler provides an additional command line option:

R instructs the Enabler to release the resources previously allocated to the DSP-100. When the 'R' option is used, any settings specified by the 'B', 'I', and 'U' options are ignored.

#### 2.2.1.1 Example 1

DSP100EN.EXE

In example 1, no command line argument is specified. The Enabler will report an error and display the proper usage of the command.

#### 2.2.1.2 Example 2

DSP100EN.EXE (s0,b300,i5)

In example 2, the Enabler will configure the DSP-100 in socket 0 with a base address of 300H and IRQ 5 using a configuration memory window at segment D000. The DSP-100's interrupt status register will be enabled.

#### 2.2.1.3 Example 3

DSP100EN.EXE (i10,u,b340,s1)

In example 3, the Enabler will configure the DSP-100 in socket 1 with a base address of 340H and IRQ 10 using a configuration memory window at segment D000. The DSP-100's interrupt status register will be disabled (Scratchpad registers enabled). Note that the parameter order is not significant.

#### 2.2.1.4 Example 4

DSP100EN.EXE (s0,b300,i3,wd8)

In example 4, the Enabler will configure the DSP-100 in socket 0 with a base address of 300H and IRQ 3 using a configuration memory window at segment D800. The DSP-100's interrupt status register will be enabled.

#### 2.2.1.5 Example 5

DSP100EN.EXE (s0,b300,i5,r)

In example 5, the Enabler will release the configuration used by the DSP-100 in socket 0 using a configuration memory window at segment D000. The base address and IRQ parameters are ignored and may be omitted.

#### 2.2.1.6 Example 6

DSP100EN.EXE (s1,r,wcc)

In example 5, the Enabler will release the configuration used by the DSP-100 in socket 1 using a configuration memory window at segment CC00.

## 2.2.2 Common Problems

### **Memory Range Exclusion:**

The Enabler requires a region of high DOS memory when configuring a DSP-100. This region is 1000H bytes (4KB) long and by default begins at address D0000H (the default address may be changed using the "W" option). If a memory manager such as EMM386, QEMM, or 386Max is installed on the system, this region of DOS memory must be excluded from the memory manager's control. Consult the documentation provided with the memory manager software for instructions on how to exclude this memory region.

Furthermore, some systems use the high memory area for BIOS shadowing to improve overall system performance. In order for the Enabler to operate, any BIOS shadowing must be disabled in the address range specified for the configuration window. BIOS shadowing can usually be disabled through the system's CMOS setup utility.

### **Socket Numbers:**

The Enabler requires the DSP-100's socket number to be specified on the command line and the DSP-100 must be inserted into the socket before the Enabler is invoked. Some vendors number their sockets from 1 to N while other vendors number their sockets from 0 to N-1. For the DSP-100 Enabler, the lowest socket number in the system is designated socket 0.

### **Card and Socket Services Software:**

In order to use the DSP-100 Enabler for DOS, the system MUST NOT be configured with Card and Socket Services software. If a Card and Socket Services software is installed, the DSP-100 Enabler may interfere with its operation and with the device(s) it controls. For systems configured with Card and Socket Services, the DSP-100 Client Driver is the recommended method of configuration.



## 3. Windows 95 Installation

---

To allow easy configuration of the DSP-100, an Windows 95 "INF" configuration file has been written for the hardware. This configuration file supports the DSP-100 in both addressing modes: block mode and "com" mode.

### 3.1 Installing a DSP-100 Under Windows 95.

1. Insert the DSP-100 into any available PC Card socket.
2. The first time a new PC Card type is installed the **New Hardware Found** window opens. After this first installation Windows 95 will automatically detect and configure the card. If the **New Hardware Found** window does not open, then skip to the next section, "DSP-100 Resource Settings".
3. The **New Hardware Found** window provides several options to configure the DSP-100 card. Click the "**Driver from Disk**" option button. Click "OK" to continue.
4. An "**Install from Disk**" dialog box should appear. Insert the diskette with the "**DSP-100.INF**" file, select the correct drive letter and path, and click "OK". Windows 95 will browse the path for the aforementioned files.
5. During the installation process, it may be required to supply the computer with the Windows 95 CD or installation diskettes. The DSP-100's serial devices will require the file "**SERIALUI.DLL**". Insert the CD or diskette and click "OK".

**IMPORTANT NOTE:**

If the user already has these files installed on the computer, or if the installation disks are unavailable, it may not be necessary to supply the computer with the Windows 95 CD or installation diskettes. If prompted for the disks, click "OK". A dialog box with an option to skip will appear. Click the "Skip" button and the files will not be installed. If these files exist in the windows system directory, those files will be used.

The DSP-100 PC Card should now be configured. In the future, Windows 95 will automatically recognize and configure the DSP-100.

## 3.2 DSP-100 Resource Settings in Windows 95

Windows 95 maintains a registry of all known hardware installed within the computer. Inside this hardware registry Windows 95 keeps track of all the computer's resources, such as base I/O addresses, IRQ levels, and DMA channels. In the case of a **PC Card (PCMCIA)** type board, Windows 95 configures the new hardware using free resources it finds within the hardware registry, and updates the registry automatically.

To view and / or edit hardware devices in Windows 95 use the system **Device Manager**. To access Device Manager double click the **System** icon in the Windows 95 control panel, or click the **My Computer** icon on the Win95 desktop with the right mouse button and select **Properties** from the pull down menu. Consult Windows 95 on-line help for details on the use of the **Device Manager**.

Windows 95 handles the DSP-100 as a "parent/child device".

- v The DSP-100 is the "parent device" and is listed under the hardware class **Multi-function Adapters** in the device manager.
- v Each serial port is a "child device" of the "parent device" DSP-100. There are 2 child COM ports for the DSP-100 which are listed under the hardware class **Ports (Com & LPT)**.

### 3.2.1 Viewing Resource Settings with Device Manager

1. Start the Windows 95 **Device Manager**.
2. Double click on the hardware class **Multi-function Adapters** to list hardware devices in the class.
3. The DSP-100 "parent device" belongs to this hardware class. The device name for the DSP-100 is **Omega Inc-PCMCIA Dual RS-232 Serial Port Card**.
4. Open the **Properties** dialog for the DSP-100 device, then click the **Resources** tab to view the Input/Output Range and Interrupt Request resource allocations. Examine and remember the Input/Output Range, then close the properties window.
5. Double click the hardware class **Ports (Com and LPT)**. Two of the Logical COM Ports (COM2, COM4, etc.) listed in this class are the "child devices" of the DSP-100 "parent device".

- View the Properties dialog for each COM port and examine the Resources allocated to each port. Inside the Resource allocation window two of the COM ports will identify the **Omega Inc-PCMCIA Dual RS-232 Serial Port Card** as the parent device. The Input/Output Range and Interrupt Request resource allocations for these two COM ports will also match the resource allocations of the DSP-100 “parent device”.

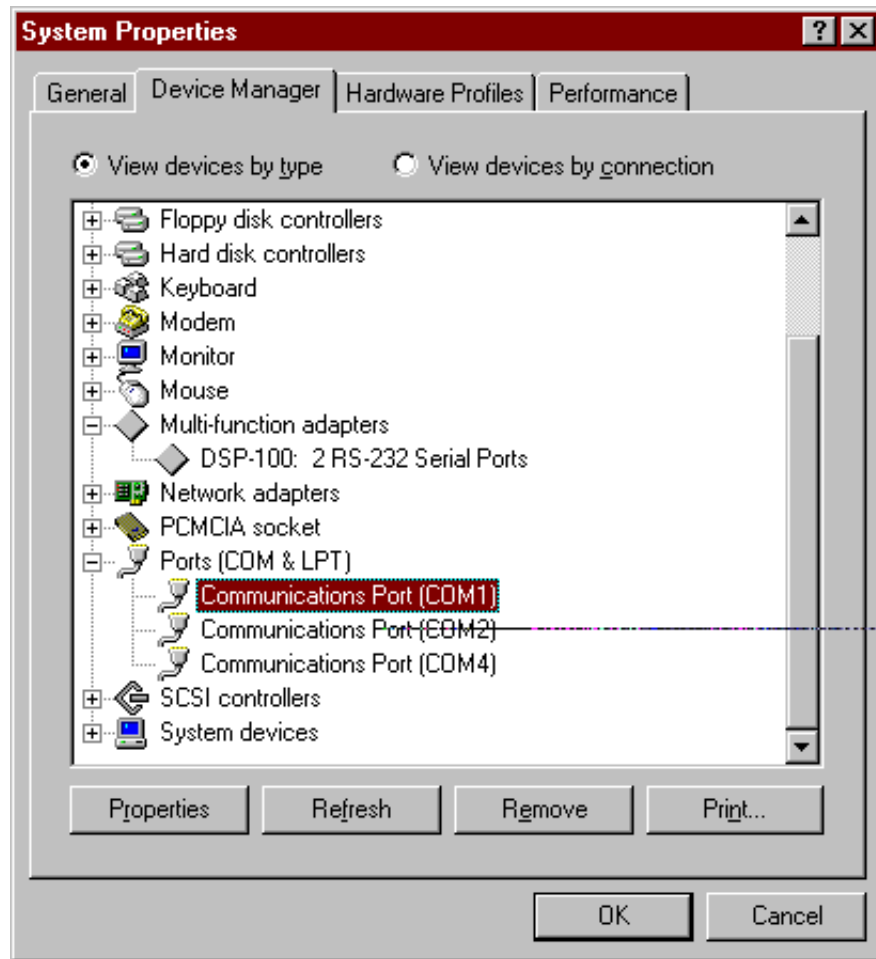


Figure 2. Windows 95 Device Manager

- Use the COM Port device names (COM2, COM4, etc.) to access any of the particular serial ports on the DSP-100. This name is required by a Windows 95 application when accessing a particular port.

### 3.2.2 Changing Resource Settings with Device Manager

The DSP-100's serial ports are addressable in two modes: block mode, and "COM" mode (see Chapter 5: Hardware Information). The DSP-100 is addressable in either mode from Windows 95.

1. Start the Windows 95 **Device Manager**.
2. Double click on the hardware class **Multi-function Adapters** to list hardware devices in the class.
3. The DSP-100 "parent device" belongs to this hardware class. The device name for the DSP-100 is **Omega Inc-PCMCIA Dual RS-232 Serial Port Card**.
4. Open the **Properties** dialog for the DSP-100 device, then click the **Resources** tab to view the Input/Output Range and Interrupt Request resource allocations.
5. Several predefined **Basic Configurations** have been included for the DSP-100. The table below defines the various configurations. When the **Use Automatic Settings** check box is enabled Windows 95 will attempt to configure the DSP-100 in the order listed.

Basic Configuration	Port Addresses		Basic Configuration	Port Addresses
0000*	3F8, 3E8		0009	220, 228
0001*	2F8, 2E8		000A	260, 268
0002	390, 398		000B	280, 288
0003	100, 108		000C	290, 298
0004	110, 118		000D	330, 338
0005	120, 128		000E	340, 348
0006	130, 138		000F	350, 358
0007	140, 148		0010	300, 308
0008	1E0, 1E8		0011	310, 318

Figure 3. DSP-100 Basic Configuration Table.

\* Indicates "COM" mode addressing. Addresses 3F8/3E8 are the standard addresses for COM1/COM3. Addresses 2F8/2E8 are the standard addresses for COM2/COM4. All other basic configurations use "block" mode addressing at non-standard base addresses. Windows 95 enumerates any COM port at a non-standard address starting with COM5.

6. Select a **Basic Configurations** that displays "No conflicts" in the bottom display region titled **Conflicting Device List** from the drop down list. Some applications may not be able to access ports higher than COM4. To use the DSP-100 PCMCIA serial ports with these applications you might be forced to remove other serial communications devices from your system.

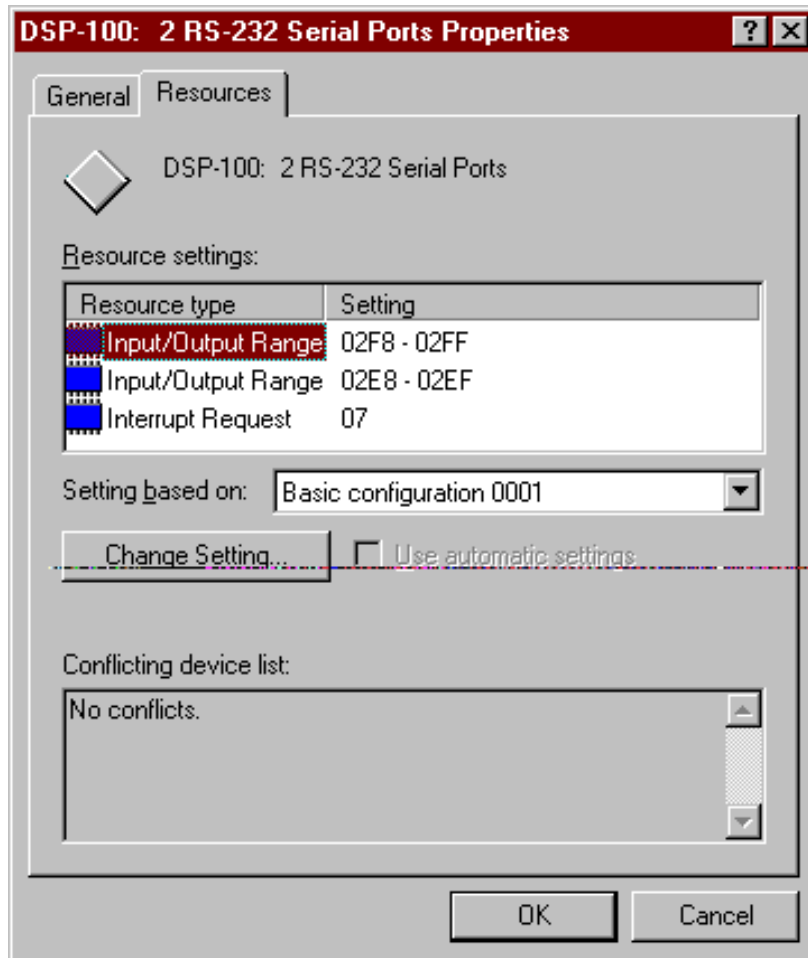


Figure 4. Windows 95 Resource Allocation

7. Windows 95 should have chosen an available **Interrupt Request** setting automatically when the I/O address range was configured by a Basic Configuration selection. This default **Interrupt Request** setting should not need changed as long as "No conflicts" is displayed in the bottom display region titled **Conflicting Device List**. If you are satisfied with Windows 95 selection then skip the next step.

8. To modify the **Interrupt Request** setting click the resource name and click the **Change Setting** button. An Edit Resource window will open up. Inside this window click on the up/down arrows to the right of the Interrupt Request value. This scrolls you through all of the allowable resources for your hardware. Pay attention to the conflict information at the bottom of the window. Do not select a value that causes a conflict with any other installed hardware.
9. If any changes have been made to the DSP-100's configuration the card will automatically be reconfigured to the new resources specified. Any time a PCMCIA card of this type is inserted Windows 95 will attempt to configure the card at these resource settings. Click the **Use Automatic Settings** box to reset this card type for automatic configuration.

## 4. OS/2 Installation

---

In order to use the DSP-100 Client Driver for OS/2, the system must be configured as follows:

1. The system must be running OS/2 version 2.1 or later.
2. OS/2 PCMCIA Card and Socket Services support must be installed. If PCMCIA support was not selected when OS/2 was installed, it can be added using the Selective Install facility in the System Setup folder. On OS/2 2.1 and 2.11, Socket Services must be added separately. The necessary files can be found on CompuServe in the OS2SUPPORT forum and may be available elsewhere. These files are not available from Omega.
3. Omega's OS/2 serial port device driver, "QCOM" version 2.01 or later, must be installed. The DSP-100 will not operate with the standard OS/2 serial port device drivers. Omega can not guarantee the operation of the DSP-100 with any other third party device drivers for OS/2.
4. There must be at least 16 bytes of available I/O space and 1 available IRQ.

After the system has been configured to the above specifications, the DSP-100 Client Driver may be installed with the following procedure:

1. Copy the DSP100.SYS client driver file from the distribution disk to any convenient directory on the hard disk.
2. Open the CONFIG.SYS file using any ASCII text editor.
3. Add the following line to the CONFIG.SYS file:

`DEVICE = drive:\path\DSP100.SYS options`  
where *options* are the DSP-100 OS/2 Client Driver command line options discussed in the following sections.

4. Save the CONFIG.SYS file, exit the text editor, shutdown the system, and reboot to activate the changes.

## 4.1 Command Line Options

The DSP-100 Client Driver for OS/2 supports two methods of configuration: using "system assigned" resources and using "user assigned" resources. Both options provide full PCMCIA compliance and functionality (including "Hot-swapping") but each has some advantages and disadvantages as discussed in the following sections.

### 4.1.1 Configuring With "System Assigned" Resources

Allowing the OS/2 Plug-and-Play system to assign the hardware resources to the DSP-100 is the ideal choice when only OS/2 programs will access the serial ports. When configuring the hardware, the user simply specifies a list of COM port numbers. When a DSP-100 is inserted into a PCMCIA socket, the client driver will configure the card as a series of COM ports, starting with the lowest available port number in the list.

Configuring a DSP-100 with system assigned resources can be a problem, however, if DOS and/or Windows applications will be accessing the serial ports. This is because most DOS applications write directly to the communications hardware and the Windows' Control Panel also wants to know the hardware configuration of the serial ports. In these cases, the user may want to configure the DSP-100 with "user assigned" resources.

#### 4.1.1.1 Example 1

```
DEVICE=C:\DSP-100\DSP100.SYS COM3
```

In example 1, the Client Driver will attempt to configure the DSP-100 as COM3 and COM4. If COM3 or COM4 already exists in the system, the DSP-100 will not be configured. Furthermore, only one DSP-100 can be installed in this system.

#### 4.1.1.2 Example 2

```
DEVICE=C:\DSP-100\DSP100.SYS COM7 COM3
```

In example 2, the Client Driver will attempt to configure the DSP-100 as COM3 and COM4. If COM3 or COM4 already exists in the system, the Client Driver will attempt to configure the DSP-100 as COM7 and COM8. If COM7 or COM8 already exist in the system, the DSP-100 will not be configured. Up to two DSP-100s can be installed in this system.



### 4.1.2 Configuring With "User Assigned" Resources

As mentioned in the previous section, allowing the OS/2 Plug-and-Play system to assign the hardware resources to the DSP-100 is ideal for OS/2 programs but can be a problem if DOS and/or Windows applications will be accessing the serial ports. This is because most DOS applications write directly to the communications hardware and the Windows' Control Panel also wants to know the hardware configuration of the serial ports. For this reason, the DSP-100 Client Driver allows the user to request specific hardware settings using a series of command line arguments of the form

(port,address,irq)

port	specifies the beginning COM port number
address	specifies the base I/O address of the DSP-100 in hexadecimal and must reside on an even 16-byte (10H) boundary.
irq	specifies the interrupt level (IRQ) of the DSP-100 in decimal. <i>irq</i> must be one of the following values: 3, 4, 5, 7, 9, 10, 11, 12, 14, or 15.

Each argument must be enclosed in parentheses and must be separated from other arguments by a space on the command line. Within each argument, the parameters must be separated using a comma (no spaces). When a DSP-100 is inserted into a PCMCIA socket, the client driver will configure the card as a series of COM ports, starting with the lowest available port number in the list.

#### **IMPORTANT:**

If the user specified resources are in-use by other devices in the system, the DSP-100 will not be configured.

#### 4.1.2.1 Example 1

```
DEVICE=C:\DSP-100\DSP100.SYS (3,100,5)
```

In example 1, the Client Driver will attempt to configure the DSP-100 as COM3 and COM4 using I/O addresses 100-10F hex and IRQ 5. If COM3 or COM4 already exists, or if the I/O address or IRQ resources are already in use, the DSP-100 will not be configured. Furthermore, only one DSP-100 can be installed in this system.

### 4.1.2.2 Example 2

```
DEVICE=C:\DSP-100\DSP100.SYS (7,120,15) (3,300,4)
```

In example 2, the Client Driver will attempt to configure the DSP-100 as COM3 and COM4 using I/O address 300-30F hex and IRQ 4. If COM3 or COM4 already exists, or if the I/O address or IRQ resources are already in use, the Client Driver will attempt to configure the DSP-100 as COM7 and COM8 using I/O address 120-12F hex and IRQ 15. If COM7 or COM8 already exists or if the I/O address or IRQ resources are already in use, the DSP-100 will not be configured. Up to two DSP-100s can be installed in this system.

### 4.1.3 Advanced Configuration Topics

For some applications, it may be desirable to specify the resources for one DSP-100 while allowing the OS/2 Plug-and-Play system to assign the hardware resources for any additional cards. This can be accomplished by mixing the configuration methods on the DSP-100 Client Driver command line

```
DEVICE=C:\DSP-100\DSP100.SYS (3,100,5) COM7
```

It is important to remember that when a DSP-100 is inserted into a PCMCIA socket, the client driver will configure the card as a series of COM ports, starting with the lowest available port number in the list.

Another common application requirement is to have a DSP-100 inserted into socket 1 be configured as COM3 and COM4 while a DSP-100 inserted into socket 2 be configured as COM7 and COM8. This type of configuration is supported by appending a "=Sx" parameter **after** any command line argument.

```
DEVICE=C:\DSP-100\DSP100.SYS COM3=S1 COM7=S2
```

```
DEVICE=C:\DSP-100\DSP100.SYS (3,100,4)=S1 (7,300,3)=S2
```

## 4.2 Monitoring The Status Of PCMCIA Cards

OS/2 Warp provides a utility called "Plug and Play for PCMCIA" that can be used to monitor the status of each PCMCIA socket. In OS/2 2.1, this utility is called "Configuration Manager". When a DSP-100 is inserted, the Card Type for the appropriate socket will display "Multi-Function". If the card is successfully configured, the Card Status will display "Ready". If the card cannot be configured, the Card Status will be "Not Ready". You can view the resources claimed by a configured card by double-clicking on that card's line in the window.

## 4.3 Common Problems

### Invalid I/O Address When Using OS/2 Version 2.1:

PCMCIA Card Services for OS/2 version 2.1 sometimes fails to supply a valid I/O address when using "system assigned" resources. Use the "Configuration Manager" program to examine the I/O address range assigned to the DSP-100. If this range does not begin on an even 16 byte (10H) boundary, the DSP-100 will have to be installed using "user assigned" resources to force a valid configuration.

There have not been any reports of this problem with OS/2 Warp.

### Resources Not Available:

When using "user assigned" resources, it is the user's responsibility to ensure the I/O address and IRQ resources are available. For OS/2 Warp users, the RMVIEW utility may be useful in finding resource conflicts. Type "rmview /?" at an OS/2 command prompt for details. OS/2 version 4.00 comes with HARDWARE MANAGER, which is easier to use than RMVIEW.

When using "system assigned" resources, if the user knows the port number is available then the system may not have sufficient resources available to configure the DSP-100. Again, the RMVIEW utility provided with OS/2 Warp may be useful in determining the problem.

Regardless of the configuration method, each command line argument specifies the first of two COM ports for the DSP-100. If any of these COM ports are already installed, the Client Driver will not load.

**Parameter Overlapping:**

When installing the DSP-100, each command line argument specifies the first of two COM ports. If these arguments overlap, the Client Driver will not load. For example, it is illegal to specify DSP100.SYS COM3 COM4 because the first argument requests COM3 - COM4 and the second argument specifies COM4 - COM5.

**Insufficient Number Of Command Line Arguments:**

The DSP-100 command line must contain at least one command line argument for each DSP-100 to be installed.

## 5. Unix Support

---

The DSP-100 and other Omega asynchronous serial communications interface products are supported in the Unix operating system by Omega's **Unix Enhanced Serial Device Driver**. This driver is **not included** with the DSP-100, but may be purchased separately.

### 5.1 Unix Enhanced Serial Device Driver

The Enhanced Serial Device Driver provides a fully interrupt driven, buffered input/output system for multiple RS-232 serial ports in an IBM or compatible microcomputer using the SCO UNIX System V Operating System. The Enhanced Serial Device Driver is a replacement for the serial driver that is provided with the SCO UNIX operating system, and is one hundred percent SCO UNIX compatible. It provides the same functions as the SCO UNIX (Release 3.2) serial device driver plus the following features:

- v Support for Omega RS-232 Asynchronous Communication Adapters, as well as, the system board serial port.
- v A flexible configuration scheme which allows any I/O address or interrupt request number that is supported by Omega's Asynchronous Communication Adapters.
- v Support for the National Semiconductor NS16550A device, an enhanced version of the industry standard NS16450 device. The NS16550A offers 16 character transmit and receive buffers that greatly reduce interrupt overhead on the CPU.
- v Support for up to 40 COM ports on multiple boards/IRQs.
- v The Enhanced Serial Device Driver will support a total of 5 serial adapters.
- v Written in C for maximum efficiency.

(This page intentionally left blank.)

## 6. Hardware Information

---

The DSP-100's two asynchronous serial ports are implemented using 2 standard 16C550 UARTs. Each of these UARTs requires 8 bytes of I/O space. The DSP-100's serial ports are addressable in two modes:

1. **'Block Mode'** : The two serial ports are configured in one 16-byte continuous block of I/O address space. The block must begin on an even 16-byte (10H) boundary (e.g. 300H, 310H, 320H, etc.).

DSP-100 RS-232 channel	Address assignment
Channel A	Base Address + 0
Channel B	Base Address + 8

Figure 5. Block Mode Addressing.

2. **'COM Mode'** : The two serial ports are configured at the standard COM port I/O address locations. The ports may be configured as COM1 and COM3; or as COM2 and COM4.

Each 16C550 UART contains 8 I/O registers. The last of these registers, located at (Base address + 7), is referred to as the 'Scratchpad Register' and provides no functionality to the UART. In place of this Scratchpad Register, the DSP-100 implements an interrupt status register which can be accessed at (Base address + 7) of any UART.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	Intr B	Intr A

Figure 6. Interrupt Status Register.

When one or more UARTs have interrupts pending, the associated bit(s) in the interrupt status register are set to logic 1. When all the pending interrupts have been serviced for a specific UART, its interrupt status bit will be cleared to logic 0 automatically. When all the pending interrupts from all UARTs have been serviced, the entire interrupt status register will return logic 0. The application program should not exit its interrupt service routine until all pending interrupts from all channels have been serviced (interrupt status register = 0) or no additional interrupts will be received.

If an application requires the UARTs' Scratchpad Registers, the interrupt status register can be disabled. Disabling the interrupt status register is supported by the DSP-200 configuration software, which is operating

system dependent. Refer to the relevant operating system installation section for specific usage of this feature.



## 7. External Connections

---

An adapter cable is included with the DSP-100 to convert the 25-pin output connector into 2 standard D-9 male RS-232 connectors as shown in the figure below.

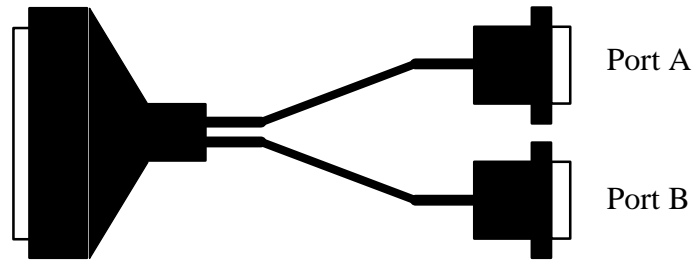


Figure 7. DSP-100 Adapter Cable.

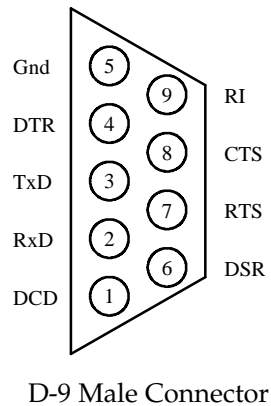


Figure 8. Standard RS-232 Signal Assignment.

(This page intentionally left blank.)

## 8. Specifications

---

<b>Bus Interface</b>	PCMCIA PC Card Standard 2.1 compliant
<b>Physical Dimensions</b>	Type II PCMCIA card (5mm)
<b>Maximum Baud Rate</b>	120K
<b>Power Requirements</b>	+5 volts    35.85 mA (typical) 45.87 mA (maximum)
<b>Connector</b>	Adapter to 2 standard male D-9