Table of Contents

1. Introduction 1-1
1.1 IOP-241 Features 1-1
1.2 System Configuration 1-1
2. DOS / Windows 3.x Installation 2-1
2.1 IOP-241 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 2-6
2.2 IOP-241 Enabler for DOS 2-7
2.2.1 Command Line Options 2-8
2.2.2 Common Problems 2-11
2.3 Where To Go From Here 2-12
3. Windows 95 Installation 3-1
3.1 Installing a IOP-241 Under Windows 95. 3-1
3.2 IOP-241 Resource Settings in Windows 95 3-2
3.2.1 Viewing Resource Settings with Device Manager 3-2
3.2.2 Changing Resource Settings with Device Manager 3-3
3.3 Where To Go From Here 3-4
4. Theory of Operation 4-1
4.1 I/O Port Description 4-1
4.2 Port C Interrupts Description 4-2
4.3 External Interrupt description 4-3
5. Register Descriptions 5-1
5.1 Data Port A Control Register (Base + 0) 5-1
5.2 Data Port B Control Register (Base + 1) 5-2
5.3 Data Port C Control Register (Base + 2) 5-2
5.4 Port C Interrupt Enable Register (Base + 5) 5-3
5.5 Interrupt Mode Control Register (Base + 6) 5-4
5.6 Interrupt Status Register (Read Only) (Base + 7) 5-5
5.7 Interrupt Acknowledge Register (Write Only) (Base + 7) 5-5
5.8 Summary of Interrupt Source Options 5-6
5.9 Summary of Input/Output Options 5-7
5.10 Programming Example 5-8
6. External Connections 6-1
7. Optional Accessories 7-1
7.1 CP-IO37 - Cable Assembly 7-1
7.2 UIO-37 - Screw Terminal Adapter 7-2
8. Specifications 8-1

List of Figures

Figure 1. IOP-241 System Configuration. 1-1

Figure 2. Client Driver versus Enabler for DOS/Windows 3.x. 2-1

Figure 3. Windows 95 Resource Settings 3-4

Figure 4. IOP-241 Program Registers 5-1

Figure 5. Data Port A Control Register 5-1

Figure 6. Data Port B Control Register 5-2

Figure 7. Data Port C Control Register 5-2

Figure 8. Port C Interrupt Control Register 5-3

Figure 9. Interrupt Mode Control Register 5-4

Figure 10. Interrupt Status Register (Read Only) 5-5

Figure 11. Interrupt Status Register (Write Only) 5-5

Figure 12. PCMCIA 33-Pin Connector 6-1

1. Introduction

The IOP-241 is a 24 channel digital input/output adapter for systems equipped with PCMCIA Type II and/or Type III expansion sockets.

1.1 IOP-241 Features

- v PC Card Standard Specifications 2.1 compliant.
- v twenty-four TTL compatible digital I/O channels.
- v channels individually programmable as either input or output.
- v eight of the I/O channels may be used as interrupt sources.
- v active high sensitive, active low sensitive, low-to-high transistion, or high-to-low transition interrupt modes.
- v external interrupt available.

1.2 System Configuration

The figure below illustrates a complete IOP-241 system. For users who do not want to interface to the IOP-241's 0.8mm I/O connector, an optional adapter cable is available to convert this connector into an industry standard D-37 female connector. For applications requiring discrete wire hook-ups, an optional screw terminal adapter is available to convert the D-37 connector into 37 discrete screw terminal blocks. These optional accessories are described in detail in chapter 7.

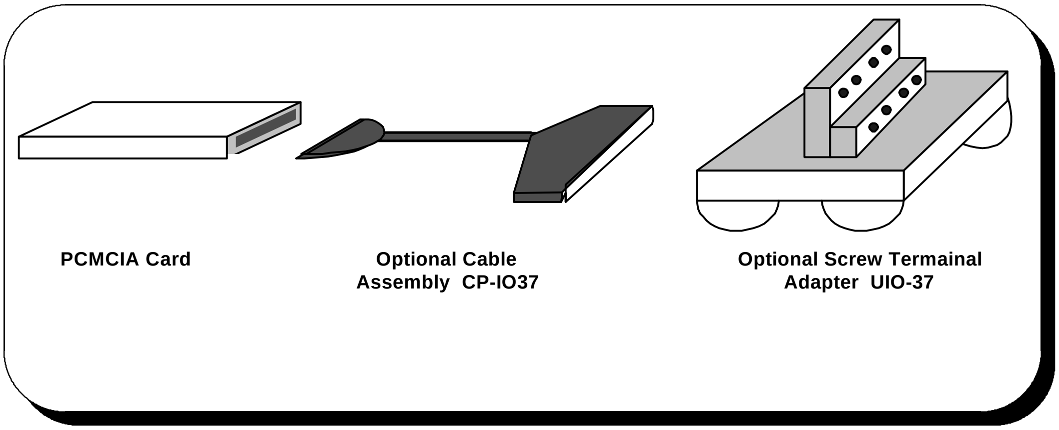

PCMCIA Card
Optional Cable Assembly CP-IO37
Optional Screw Termainal Adapter UIO-37

Figure 1. IOP-241 System Configuration.

(This page intentionally left blank.)

2. DOS / Windows 3.x Installation

Two configuration software programs are provided with the IOP-241: a Client Driver, IOP241CL.SYS, and a card Enabler, IOP241EN.EXE. Both of these programs are executed from DOS (before entering Windows) and allow operation of the IOP-241 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: IOP241CL.SYS	File name: IOP241EN.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 IOP-241 adapters upon insertion (Hot Swapping)	Does not support automatic configuration of IOP-241 adapters upon insertion (Hot Swapping)
Requires PCMCIA Card and Socket Services software	Does not require PCMCIA Card and Socket Services software

Figure 2. 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 IOP-241 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 IOP-241 Client Driver for DOS

In order to use the IOP-241 Client Driver, the system must be configured with Card and Socket Services software. Card and Socket Services software is not provided with the IOP-241 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 IOP-241 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.

2.1.1 Client Driver Installation

The following procedure is used to install the IOP-241 Client Driver:

1. Copy the file IOP241CL.SYS from the IOP-241 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\\IOP241CL.SYS *options*

   where *options* are the IOP-241 Client Driver command line options discussed on the following pages.
5. Save the CONFIG.SYS file and exit the text editor.
6. Insert the IOP-241 into one of the system's PCMCIA slots.

   NOTE: Since the IOP-241 Client Driver supports "Hot Swapping", it is not necessary to have the IOP-241 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 IOP-241 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 IOP-241 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 IOP-241 may be removed and / or inserted from the system as desired. On each insertion into the PCMCIA socket, the IOP-241 will be automatically reconfigured according to the command line options.

2.1.2 Command Line Options

The IOP-241 Client Driver accepts up to eight command line arguments from the user to determine the configuration of the IOP-241. If any arguments are provided, the Client Driver will attempt to configure any IOP-241s 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 <u>space</u> on the command line. Within each argument, any or all of the following parameters may be specified using a <u>comma</u> (no spaces) to separate each parameter:

B*address* specifies the base I/O address of the IOP-241 in hexadecimal. This address must reside on an even 8-byte boundary (I/O base address must end in '0' or '8'). The valid range for the IOP-241's base address is 100H to 3F8H. If this option is omitted, a base address will be assigned by Card and Socket Services.

I*irq* specifies the interrupt level (IRQ) of the IOP-241 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.

S*socket* specifies which PCMCIA socket the IOP-241 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 IOP-241s inserted into any socket.

2.1.2.1 Example 1

```
DEVICE = C:\IOP-241\IOP241CL.SYS
```

In example 1, no command line arguments are specified. The Client Driver will configure a IOP-241 inserted into any socket with a base address and IRQ assigned by Card and Socket Services.

2.1.2.2 Example 2

```
DEVICE = C:\IOP-241\IOP241CL.SYS (b330)
```

In example 2, a single command line argument is provided. The Client Driver will attempt to configure a IOP-241 inserted into any socket at address 330H and an IRQ assigned by Card and Socket Services. If address 330H is unavailable, the IOP-241 will not be configured.

2.1.2.3 Example 3

```
DEVICE = C:\IOP-241\IOP241CL.SYS (s0,b300,i5)
```

In example 3, a single command line argument is provided. The Client Driver will attempt to configure a IOP-241 inserted into socket 0 with a base address of 300H and IRQ 5. If address 300H or IRQ 5 is unavailable, the IOP-241 will not be configured. In addition, if a IOP-241 is inserted into any other socket, it will not be configured.

2.1.2.4 Example 4

```
DEVICE = C:\IOP-241\IOP241CL.SYS (i11,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 IOP-241 inserted into any socket with a base address of 300H and IRQ 11. If address 300H or IRQ 11 is unavailable, the IOP-241 will not be configured.

2.1.2.5 Example 5

```
DEVICE = C:\IOP-241\IOP241CL.SYS (b300,i5) (i10) ( )
```

In example 5, three command line arguments are provided. The Client Driver will first attempt to configure a IOP-241 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 IOP-241 with a base address and an IRQ assigned by Card and Socket Services.

2.1.2.6 Example 6

```
DEVICE = C:\IOP-241\IOP241CL.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 IOP-241 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:\IOP-241\IOP241CL.SYS (s0,b300,i5) (s1,b340,i10)
```

The type of configuration shown in example 7 may be desirable in systems where more than one IOP-241 is to be installed. In this example, the Client Driver will attempt to configure a IOP-241 inserted into socket 0 with a base address of 300H and IRQ 5. If the IOP-241 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 IOP-241'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 IOP-241 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 IOP-241 causing the IOP-241 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 IOP-241 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 IOP-241 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 IOP-241 Enabler for DOS

For systems that are not operating PCMCIA Card and Socket Services software, the IOP-241 DOS Enabler may be used to enable and configure the adapter. This Enabler, IOP241EN.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 IOP-241 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 IOP-241 Enabler may interfere with its operation and with the device(s) it controls.

The IOP-241 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 IOP241EN.EXE. If more than one adapter is installed in a system, the Enabler must be executed separately for each adapter. Furthermore, IOP241EN.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 IOP-241. 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 IOP-241 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:

S*socket* specifies which PCMCIA socket the IOP-241 must be inserted into for this configuration argument to be used. *socket* must be in the range 0 - 15. This option is always required.

B*address* specifies the base I/O address of the IOP-241 in hexadecimal. This address must reside on an even 8-byte boundary (I/O base address must end in '0' or '8'). The valid range for the IOP-241's base address is 100H to 3F8H. Specify only **one** of the following options: B*address* or 'R'. Use of one of these options is **always** required.

I*irq* specifies the interrupt level (IRQ) of the IOP-241 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.

W*address* specifies the base address of the memory window required to configure the IOP-241. 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.

Before removing a IOP-241 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 on additional command line option:

R instructs the Client Driver to release the resources previously allocated to the IOP-241. When the 'R' option is used, any settings specified by the 'B', 'I', 'U', and 'E' options are ignored. Specify only **one** of the following options: B*address*, or 'R'. Use of one of these options is **always** required.

2.2.1.1 Example 1

```
IOP241EN.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

```
IOP241EN.EXE (s0,b300,i5)
```

In example 2, the Enabler will configure the IOP-241 in socket 0 with a base address of 300H and IRQ 5 using a configuration memory window at segment D000.

2.2.1.3 Example 3

```
IOP241EN.EXE (i10,b340,s1)
```

In example 3, the Enabler will configure the IOP-241 in socket 1 with a base address of 340H and IRQ 10 using a configuration memory window at segment D000. Note that the parameter order is not significant.

2.2.1.4 Example 4

```
IOP241EN.EXE (s0,b300,i3,wd8)
```

In example 4, the Enabler will configure the IOP-241 in socket 0 with a base address of 300H and IRQ 3 using a configuration memory window at segment D800.

2.2.1.5 Example 5

```
IOP241EN.EXE (s0,b300,i5,r)
```

In example 5, the Enabler will release the configuration used by the IOP-241 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

```
IOP241EN.EXE (s1,r,wcc)
```

In example 5, the Enabler will release the configuration used by the IOP-241 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 IOP-241. 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 IOP-241's socket number to be specified on the command line and the IOP-241 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 IOP-241 Enabler, the lowest socket number in the system is designated socket 0.

Card and Socket Services Software:
In order to use the IOP-241 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 IOP-241 Enabler may interfere with its operation and with the device(s) it controls. For systems configured with Card and Socket Services, the IOP-241 Client Driver is the recommended method of configuration.

2.3 Where To Go From Here

The IOP-241 is now configured and ready for use. Depending on the type of application software to be used, the user may wish to review one or more of the following:

1. Chapter 4 of this document provides a basic theory of operation of the adapter for users who wish to learn the technical details about the operation of the IOP-241.
2. For users who want to program the adapter with direct I/O transfers to the IOP-241's register set, chapter 5 provides an address map and a detailed description of each I/O register.
3. Users who would like to write custom application software without programming the IOP-241 directly should consult the DAQDRIVE software reference manual. DAQDRIVE provides a library of data acquisition subroutines and is included free of charge with the IOP-241.
4. For turn-key data acquisition software (i.e. LabTech Notebook or SnapMaster) consult the documentation provided by the software manufacturer.

(This page intentionally left blank.)

3. Windows 95 Installation

To allow easy configuration of the IOP-241, an Windows 95 **"INF"** configuration file has been written for the hardware.

3.1 Installing a IOP-241 Under Windows 95.

1. Insert the IOP-241 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, “IOP-241 Resource Settings".
3. The **New Hardware Found** window provides several options to configure the IOP-241 card. Click the **"Driver from Disk"** option button. Click "OK" to continue.
4. An **"Install from Disk"** dialog box should appear. Insert the IOP-241 customer software diskette, select the correct drive letter and path for the configuration file (name will have .INF extension), and click "OK". Windows 95 will browse the path for the aforementioned files.

The IOP-241 PC Card should now be configured. In the future, Windows 95 will automatically recognize and configure the IOP-241.

3.2 IOP-241 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.

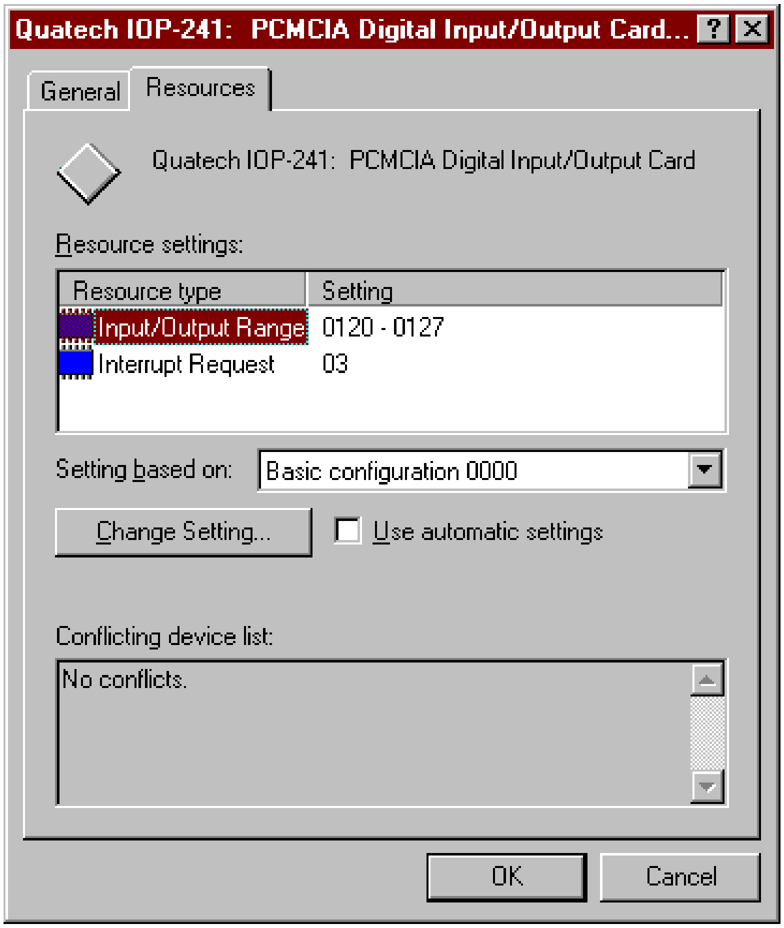
To view and / or edit hardware devices in Windows 95 use the system **Device Manager**. To access Device Manager double click the **System** icon in the Windows 95 control panel, or click the **My Computer** icon on the Windows 95 desktop with the right mouse button and select **Properties** from the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

3.2.1 Viewing Resource Settings with Device Manager

1. Start the Windows 95 **Device Manager**.
2. Double click on the hardware class **Data Acquisition** to list hardware devices in the class.
3. The IOP-241 belongs to this hardware class. The device name for the IOP-241 is **Omega IOP-241: PCMCIA Digital Input/Output Card.**
4. Open the **Properties** dialog for the IOP-241 device, then click the **Resources** tab to view the Input/Output Range and Interrupt Request resource allocations (see Figure 3).
5. To access the IOP-241 use these system resources allocated by Windows 95 or see **Changing Resource Settings with Device Manager**.

3.2.2 Changing Resource Settings with Device Manager

1. Start the Windows 95 **Device Manager**.
2. Double click on the hardware class **Data Acquisition** to list hardware devices in the class.
3. The IOP-241 belongs to this hardware class. The device name for the IOP-241 is **Omega IOP-241: PCMCIA Digital Input/Output Card.**
4. Open the **Properties** dialog for the IOP-241 device, then click the **Resources** tab to view the Input/Output Range and Interrupt Request resource allocations.



Quatech IOP-241: PCMCIA Digital Input/Output Card

Resource settings:

Resource type	Setting
Input/Output Range	0120 - 0127
Interrupt Request	03

Setting based on: Basic configuration 0000

□ Use automatic settings

Conflicting device list:

No conflicts.

Figure 3 Windows 95 Resource Settings

5. To modify either of the resource settings click the resource name and click the **Change Setting** button.

6. An **Edit Resource** window will open up. Inside these Edit Resource windows click on the up/down arrows to the right of the resource 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 resource that causes a conflict with any other installed hardware.
7. Repeat the above steps to modify all of the resources allocated to the IOP-241. Once satisfied with the settings make a note of the new settings and click the OK button to accept. Clicking the Cancel button does not save your changes.
8. If any changes have been made to the IOP-241′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 (see Figure 3).

3.3 Where To Go From Here

The IOP-241 is now configured and ready for use. Depending on the type of application software to be used, the user may wish to review one or more of the following:

1. Chapter 4 of this document provides a basic theory of operation of the adapter for users who wish to learn the technical details about the operation of the IOP-241.
2. For users who want to program the adapter with direct I/O transfers to the IOP-241's register set, chapter 5 provides an address map and a detailed description of each I/O register.
3. Users who would like to write custom application software without programming the IOP-241 directly should consult the DAQDRIVE software reference manual. DAQDRIVE provides a library of data acquisition subroutines and is included free of charge with the IOP-241.
4. For turn-key data acquisition software (i.e. LabTech Notebook or SnapMaster) consult the documentation provided by the software manufacturer.

4. Theory of Operation

4.1 I/O Port Description

The 24 digital I/O channels which are provided in the IOP-241 are grouped into three different ports with each port containing eight digital I/O channels. These three ports are controlled via the Data Port A Control Register, Data Port B Control Register, and Data Port C Control Register. In each of these three registers, each bit corresponds to one data line. The Data Port A Control Register is used to access data lines DATA7 - DATA0. The Data Port B Control Register is used to access data lines DATA15 - DATA8. The Data Port C Control Register is used to access data lines DATA23 - DATA16.

Each of the 24 I/O channels (DATA23 - DATA0) may be individually programmed for either input or output. Each data port is latched on a write to that particular port. Each I/O channel may be programmed for input by writing a '1' to the appropriate bit in the corresponding data port control register. Each I/O channel is driven by an open-collector driver; therefore, writing a '1' will turn the open-collector output driver 'off'. When 'off', the open-collector output driver is tri-stated. Tri-stating the output driver of an I/O channel will allow that I/O channel to be driven by another device. An I/O channel which is configured for input may be accessed by reading the appropriate data port control register. If a '0' is latched on a particular I/O channel by writing a '0' to the appropriate bit in the corresponding data port control register , the channel may not be used as input. The channel will be 'masked' and a '0' will always be read on the channel.

If an I/O channel is to be used as output, the appropriate data port control register should be written. When a write operation is performed, the data is latched. If a bit is written with a '0', the open-collector output driver will be turned 'on' and the I/O channel will be driven to a low voltage state. Writing a '1' to the appropriate bit will turn the open-collector driver 'off', but a pull-up resistor will pull the particular I/O channel to a TTL high voltage level.

Upon reset of the IOP-241, the three data port control registers are all latched with a value of FFh. This forces all 24 open-collector output drivers 'off' upon a system power-up or an IOP-241 card insertion. Thus, all 24 I/O channels are configured as input by default. Care should be taken prior to programming any of these I/O channels for output. The open-collector drivers should not be turned 'on' (by writing a '0' to the appropriate bit of a data port control register) if the I/O channel is driven by another device. Driving the channel low by latching a '0' into a bit of the data port control register may cause damage to the peripheral, the host system, or the IOP-241 card if the channel is also driven by a peripheral.

4.2 Port C Interrupts Description

The eight Port C I/O channels (DATA23 - DATA16) may also be configured as interrupt sources. If any of these eight I/O channels is to be used to generate an interrupt, the I/O channel must be configured for input by latching the appropriate bit in the Data Port C Control Register to '1'. Also, the interrupt must be enabled by setting the appropriate bit in the Port C Interrupt Enable Register. Eight interrupt sources (INT7 - INT0) may be enabled in this manner; each of these interrupt sources corresponds to an I/O channel in Port C.

Also, the mode of the port C interrupt sources may be configured in one of four possible manners: level sensitive active low interrupt, level sensitive active high interrupt, high-to-low transition edge sensitive interrupt, and low-to-high transition edge sensitive interrupt. The lower nibble (4 bits) and upper nibble (4 bits) of the Port C interrupt sources may be configured separately. This will allow INT7 - INT4 to be configured for a different mode than INT3 - INT0. These modes are configured by writing the Interrupt Mode Control Register.

Whenever an interrupt is generated due to a Port C interrupt source, the corresponding bit of the Interrupt Status Register is set to reflect the cause of the interrupt. This provides a mechanism for determining the source of a detected interrupt. The Interrupt Status Register will be continually updated as additional interrupt generating conditions appear.

Writing a '1' to the appropriate bit of the Interrupt Acknowledge Register is the method by which interrupts should be acknowledged. After a write to the Interrupt Acknowledge Register, another interrupt will be generated if the Interrupt Status Register does not contain a value of 00h. Any bit in the Interrupt Status Register which has a value of '1' can be reset to a value of '0' if the following two conditions are met: first, the corresponding bit in the Interrupt Acknowledge Register must be written with a '1', and second, the interrupt generating condition must no longer exist. For level sensitive interrupts, an interrupt will be immediately generated after the write of the Interrupt Acknowledge Register if the interrupt generating condition (active level on Port C interrupt source) remains.

4.3 External Interrupt description

In addition to the eight Port C interrupt sources, an additional external interrupt source is provided in the IOP-241. This external interrupt source is accessed through Pin 29 of the external connector. The external interrupt source permits the IOP-241 to be operated with 24-bit input/output and one separate interrupt source.

The Interrupt Mode Control Register provides a means of enabling/disabling this external interrupt, setting the external interrupt mode, reading the status of the external interrupt, and acknowledging the external interrupt. The functionality of this external interrupt source is identical to that described for the Port C interrupt sources in the previous section.

(This page intentionally left blank.)

5. Register Descriptions

The program registers of the IOP-241 occupy eight contiguous bytes of I/O address space. These registers must be programmed to control the operation of the IOP-241. The table below lists the program registers along with their offsets relative to the I/O space base address at which the IOP-241 is located:

Offset	Read/Write	Register
0	R/W	Data Port A Control Register
1	R/W	Data Port B Control Register
2	R/W	Data Port C Control Register
5	R/W	Port C Interrupt Enable Register
6	R/W	Interrupt Mode Control Register
7	R	Interrupt Status Register
7	W	Interrupt Acknowledge Register

Figure 4. IOP-241 Program Registers

Each register of the IOP-241 is discussed in detail in the following sections.

5.1 Data Port A Control Register (Base + 0)

The Data Port A Control Register contains the control bits for I/O channels DATA0 - DATA7. Each I/O channel may be individually programmed for input by writing a '1' to the appropriate bit of this register.

Bit	Name	Description
7	DATA7	General Purpose I/O bit 7
6	DATA6	General Purpose I/O bit 6
5	DATA5	General Purpose I/O bit 5
4	DATA4	General Purpose I/O bit 4
3	DATA3	General Purpose I/O bit 3
2	DATA2	General Purpose I/O bit 2
1	DATA1	General Purpose I/O bit 1
0	DATA0	General Purpose I/O bit 0

Figure 5. Data Port A Control Register

5.2 Data Port B Control Register (Base + 1)

The Data Port B Control Register contains the control bits for I/O channels DATA8 - DATA15. Each I/O channel may be individually programmed for input by writing a '1' to the appropriate bit of this register.

Bit	Name	Description
7	DATA15	General Purpose I/O bit 15
6	DATA14	General Purpose I/O bit 14
5	DATA13	General Purpose I/O bit 13
4	DATA12	General Purpose I/O bit 12
3	DATA11	General Purpose I/O bit 11
2	DATA10	General Purpose I/O bit 10
1	DATA9	General Purpose I/O bit 9
0	DATA8	General Purpose I/O bit 8

Figure 6. Data Port B Control Register

5.3 Data Port C Control Register (Base + 2)

The Data Port C Control Register contains the control bits for I/O channels DATA16 - DATA23. Each I/O channel may be individually programmed for input by writing a '1' to the appropriate bit of this register. In addition, any of these eight I/O channels which are programmed for input may also be used to generate interrupts. In order to program an I/O channel as an interrupt source, the Port C Interrupt Enable Register must be set appropriately. Also, the I/O channel must be programmed for input by writing a '1' to the appropriate bit of the Data Port C Control Register.

Bit	Name	Description
7	DATA23/INT7	General Purpose I/O bit 23 and Interrupt Line 7
6	DATA22/INT6	General Purpose I/O bit 22 and Interrupt Line 6
5	DATA21/INT5	General Purpose I/O bit 21 and Interrupt Line 5
4	DATA20/INT4	General Purpose I/O bit 20 and Interrupt Line 4
3	DATA19/INT3	General Purpose I/O bit 19 and Interrupt Line 3
2	DATA18/INT2	General Purpose I/O bit 18 and Interrupt Line 2
1	DATA17/INT1	General Purpose I/O bit 17 and Interrupt Line 1
0	DATA16/INT0	General Purpose I/O bit 16 and Interrupt Line 0

Figure 7. Data Port C Control Register

5.4 Port C Interrupt Enable Register (Base + 5)

INT7 - INT0 may be enabled by writing the appropriate bits in the Port C Interrupt Enable Register. The corresponding I/O channel must be configured as an input channel via the Data Port C Control Register if an interrupt is to be generated. If an I/O channel is configured as an interrupt source (INT7 - INT0), the I/O channel continues to be a standard data input channel (DATA23 - DATA16) and may be read just as any other input signal is read.

Bit	Name	Description
7	INT7EN	1 = Enables INT7; 0 = Disables INT7
6	INT6EN	1 = Enables INT6; 0 = Disables INT6
5	INT5EN	1 = Enables INT5; 0 = Disables INT5
4	INT4EN	1 = Enables INT4; 0 = Disables INT4
3	INT3EN	1 = Enables INT3; 0 = Disables INT3
2	INT2EN	1 = Enables INT2; 0 = Disables INT2
1	INT2EN	1 = Enables INT1; 0 = Disables INT1
0	INT1EN	1 = Enables INT0; 0 = Disables INT0

Figure 8. Port C Interrupt Control Register

5.5 Interrupt Mode Control Register (Base + 6)

The mode of both the external interrupt and the Port C interrupts may be controlled with the Interrupt Mode Control Register. The upper nibble (4 bits) and lower nibble (4 bits) of the Port C interrupts may be configured separately. Also, the external interrupt may be enabled by writing this register. The status of the external interrupt may be read in this register, and the external interrupt may also be acknowledged by writing the appropriate bit in the Interrupt Mode Control Register.

Bit	Name	Description

7	ExtIntStat(Read)
	ExtIntAck(Write)
		ExtIntStat: to read status of external interrupt
		ExtIntAck: write acknowledges external interrupt

6	ExtraIntEn	1 = external interrupt is enabled 0 = external interrupt is disabled
5:4	ExtIntControl	These two bits control the mode of the external interrupt: 00 = Level Sensitive Active Low Interrupt 01 = Level Sensitive Active High Interrupt 10 = High-to-low Transition Edge Sensitive Interrupt 11 = Low-to-high Transition Edge Sensitive Interrupt
3:2	UpperIntCntrl	These two bits control the mode of the Upper Nibble of Port C (INT7, INT6, INT5, INT4): 00 = Level Sensitive Active Low Interrupt 01 = Level Sensitive Active High Interrupt 10 = High-to-low Transition Edge Sensitive Interrupt 11 = Low-to-high Transition Edge Sensitive Interrupt
1:0	LowerIntCntrl	These two bits control the mode of the Lower Nibble of Port C(INT3, INT2, INT1, INT0): 00 = Level Sensitive Active Low Interrupt 01 = Level Sensitive Active High Interrupt 10 = High-to-low Transition Edge Sensitive Interrupt 11 = Low-to-high Transition Edge Sensitive Interrupt

Figure 9. Interrupt Mode Control Register

5.6 Interrupt Status Register (Read Only) (Base + 7)

On a read, this register provides the interrupt status for the Port C interrupts. This provides a mechanism for determining the sources of any pending interrupts. A '1' signals that an interrupt generating condition has occurred on the appropriate channel. Interrupts will continue to occur until this register has a value of 00h and no interrupt generating conditions remain. This register must be 'reset' by acknowledging interrupts via writing the Interrupt Acknowledge Register.

Bit	Name	Description
7:0	ChanCIntStatus	The Status of INT7 - INT0 is read (Bit 7 = INT7, Bit 6 = INT6, etc...)

Figure 10. Interrupt Status Register (Read Only)

5.7 Interrupt Acknowledge Register (Write Only) (Base + 7)

Writing a '1' to any bit in the Interrupt Acknowledge Register will acknowledge the interrupt generating condition which was represented in the corresponding bit of the Interrupt Status Register. If a '1' is written to a bit in the Interrupt Acknowledge Register and the corresponding interrupt generating condition is not present, then the appropriate bit in the Interrupt Status Register will be reset (set to '0').

Bit	Name	Description
7:0	ChanCIntAck	Written to acknowledge INT 7- INT0 (Bit 7 = INT7, Bit 6 = INT6,etc...)

Figure 11. Interrupt Status Register (Write Only)

5.8 Summary of Interrupt Source Options

Two interrupt source options are provided in the IOP-241 and are summarized below:

(1) Port C Interrupt Sources

- (a) The following must be programmed:
    - (i) Channel set for input via Data Port C Control Register
    - (ii) Interrupt source enabled via Port C Interrupt Enable Register
    - (iii) Mode selected via Interrupt Mode Control Register
- (b) Interrupt generated unless Interrupt Status Register is 00h.
    - (ii) Unacknowledged interrupts are represented by a '1' in the Interrupt Status Register.
    - (i) Interrupt is generated after write of Interrupt Acknowledge Register if any interrupts remain unacknowledged.
- (c) '1' in Interrupt Status Register is reset to '0' if the following two requirements are satisfied:
    - (i) Interrupt acknowledged by writing appropriate bit in Interrupt Acknowledge Register with '1'
    - (ii) Condition which caused interrupt is no longer present.

(2) External Interrupt Source

- (a) The following must be programmed:
    - (i) Interrupt source enabled via Interrupt Mode Control Register
    - (ii) Mode selected via Interrupt Mode Control Register
- (b) Interrupt generated if ExtIntStat of Interrupt Mode Control Register is '1'
- (c) '1' in ExtIntStatus is reset to '0' if the following two requirements are satisfied:
    - (i) Interrupt acknowledged by writing ExtIntAck with '1' .
    - (ii) Condition which caused interrupt is no longer present

5.9 Summary of Input/Output Options

Each of the three ports (Port A, Port B, Port C) may be configured in one of three manners:

(1) PORT USED AS AN OUTPUT

(a) Write a '1' to the appropriate bits of the latch in order to turn 'off' the output module.

(b) Write a '0' to the appropriate bits of the latch in order to turn 'on' the output module.

(2) PORT USED AS AN INPUT

(a) Writing a '1' to all the bits of the latch will allow all the channels of the port to be read as inputs.
(Writing a '0' to any bits of the latch will 'mask' those bits. Those bits will always be read as a '0'.)

(3) PORT USED AS INPUT AND OUTPUT

If a port is to be used so that some channels are inputs and some channels are outputs, it must be insured that the channels to be used as inputs are initialized as inputs. This is done by writing a '1' to the bits of the data port control register representing the I/O channels which are to be used as inputs anytime the port is written.

(a) The channels which are to be inputs should always be written with a '1' and never written with a '0'.

(b) The channels which are to be outputs should be written with the appropriate value ('0' or '1').

(c) The channels which are latched with a '0' will always be '0' when read (they are 'masked' from input).

5.10 Programming Example

The following C program segment demonstrates how to program an IOP-241 located at I/O base address 300h. Port A will be programmed as output, and 55h will be latched at Port A. Port B will be configured with it's upper 3 bits as output and it's lower five bits as input. The upper three bits will be latched with 010b. Port C will be configured as input. Interrupts will be enabled as explained in the program comments.

```
            outp(0x300,0x55);         /* writes 55h to Port A */

            outp(0x301,0x5F);         /* writes 010b to Port B upper 3 bits*/

                                      /* sets Port B  lower 5 bits for input */

            outp(0x302,0xFF);         /* initializes Port C for input */

            Port_B = inp(0x301);      /* reads data from Port B */


/* Port_B(bit7) = '0' and  Port_B(bit5) = '0' due to 'masking' */


            Port_C = inp(0x302);      /* reads data from Port C */

            outp(0x306,0x76);

            /* INT7 - INT4 set for Level Sensitive Active
               High Interrupt Mode */

            /* INT3 - INT0 set for Edge Sensitive High-to-low Transition
               Interrupt Source */

            /* External Interrupt Source Enabled */

            /* External Interrupt Source set for Low-to-high Edge
               Sensitive Interrupt Source*/

            outp(0x305,0x77);

            /* Enables six Port C interrupt sources */

            /* Enables INT6,INT5,INT4,INT2,INT1,INT0 */
```

6. External Connections

The IOP-241 is fitted with a 33-pin 0.8mm shielded connector with the pins assigned as shown in the figure below. A mating connector is available from AMP (order part number 558126-4).

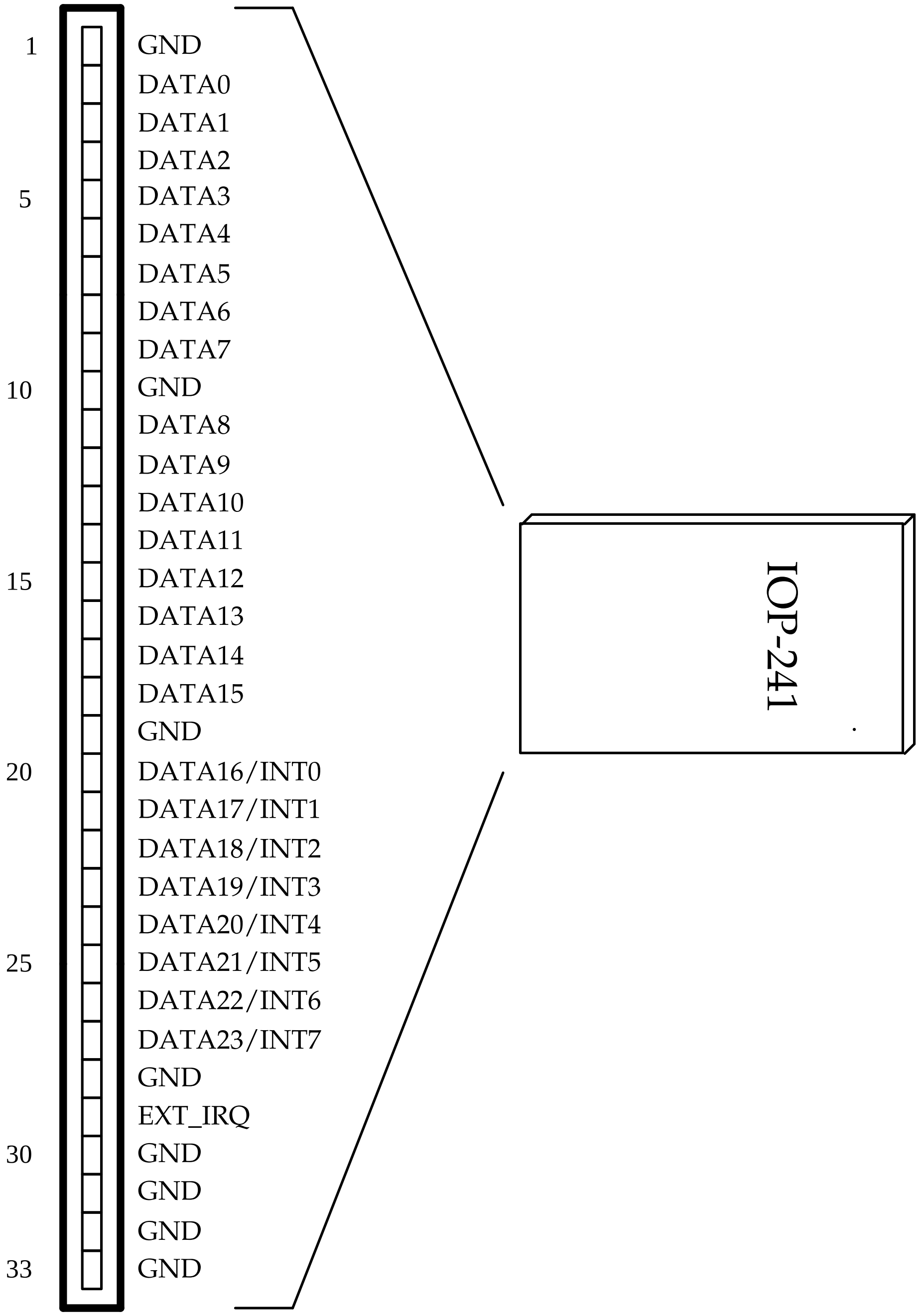


Figure 12. PCMCIA 33-Pin Connector

(This page intentionally left blank.)

7. Optional Accessories

7.1 CP-IO37 - Cable Assembly

An optional cable assembly, Omega part number CP-IO37, is available to convert the IOP-241's 33-pin 0.8mm I/O connector to a standard D-37 male connector. The first 31 connections on the IOP-241 map directly to the first 31 pins of the D37 connector. Note that two of the IOP-241's ground connections (pins 32 and 33) are not available when using the CP-IO37.

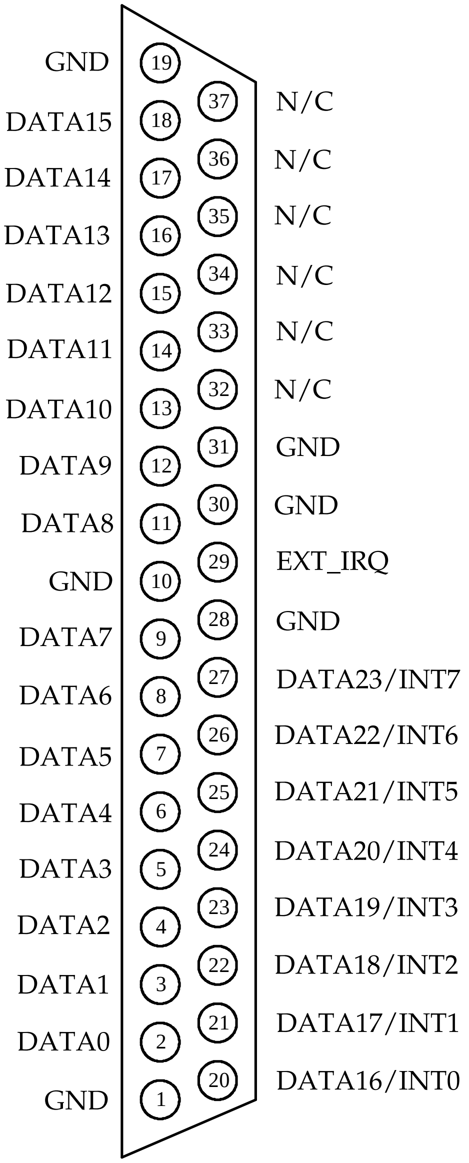


Figure 13. IOP-241 output connections using the optional CP-IO37.

7.2 UIO-37 - Screw Terminal Adapter

The UIO-37 Screw Terminal Adapter connects directly to the optional CP-IO37 cable assembly to provide a screw terminal interface to users of the IOP-241. The 37 pins of the CP-IO37 connect directly to the 37 screw terminal blocks of the UIO-37. Each screw terminal is numbered for easy reference.

NOTE:

> Since only the first 31 connections on the IOP-241 are available on the CP-IO37 cable assembly, two of the IOP-241's ground connections (pins 32 and 33) are not available when using the UIO-37.

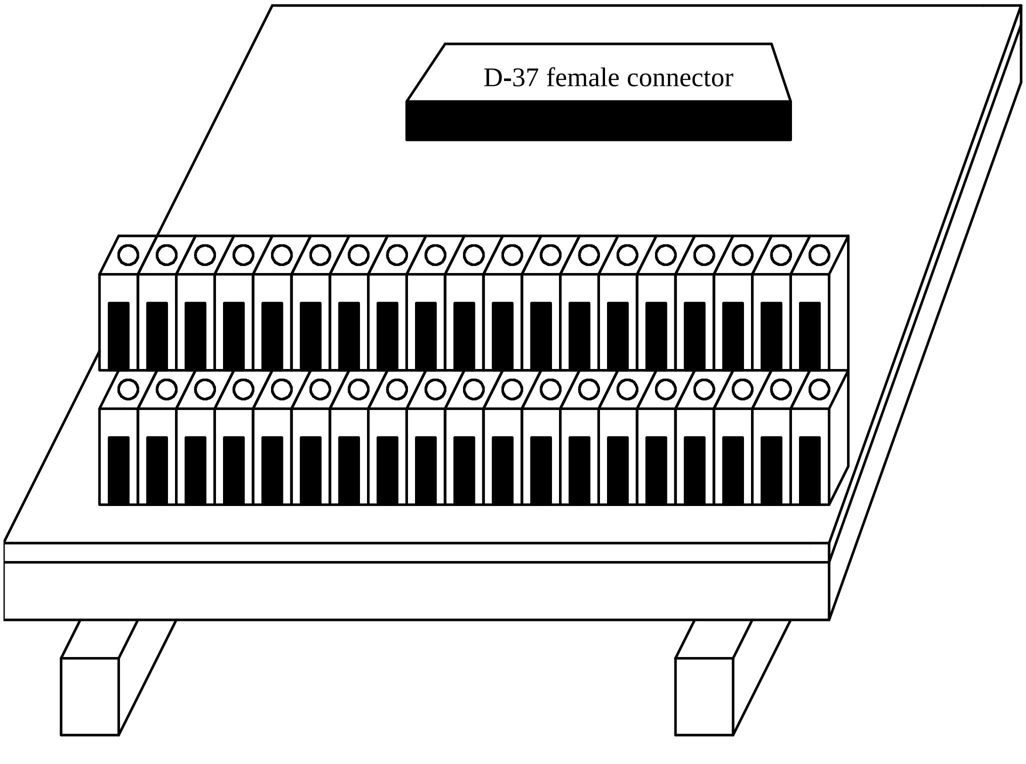


Figure 14. UIO-37 screw terminal adapter.

8. Specifications

Bus Interface	PCMCIA PC Card Standard 2.1 compliant
Physical Dimensions	Type II PCMCIA card (5mm)
Maximum Baud Rate	120K
Power Requirements	+5 volts 7.33 mA Typical (all outputs 'off') 12.38 mA Maximum (all outputs 'off') 36.38 mA Maximum (all outputs 'on')
Digital Input/Output	TTL Compatible
Current Source/Sink (at 25 C)	Sink 6mA(min) at 0.33V Sink 20mA(min) at 1.0V Source 6mA(min) at 4.0V Source 20mA(min) at 2.0V
Input/Output Current	25mA Maximum
Connector	Adapter to standard male D-37