Table of Contents

List of Figures

1. Introduction

The Omega DA8P-12 is an 8 channel analog output adapter for systems equipped with a type II PCMCIA slot. Two versions of the adapter are available:

DA8P-12U
8 independent unipolar analog outputs
output voltages from 0 to +5 volts in 1.22mV increments

DA8P-12B
8 independent bipolar analog outputs
output voltages from -5 to +5 volts in 2.44mV increments

Features common to both versions of the adapter include:

- 12-bit resolution
- Less than 10µs settling time to ½ LSB
- 1mA output current on each analog channel
- simultaneous output on 2 to 8 channels
- 8 bits of digital I/O individually programmable as input or output
- on-board event timer to control output data rates
- external interrupt input

NOTE:

Throughout this document, any reference to the DA8P-12 refers to information common to both the DA8P-12U and the DA8P-12B.

Figure 1 below illustrates a complete DA8P-12 system. For users who do not want to interface to the DA8P-12'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 8.

.

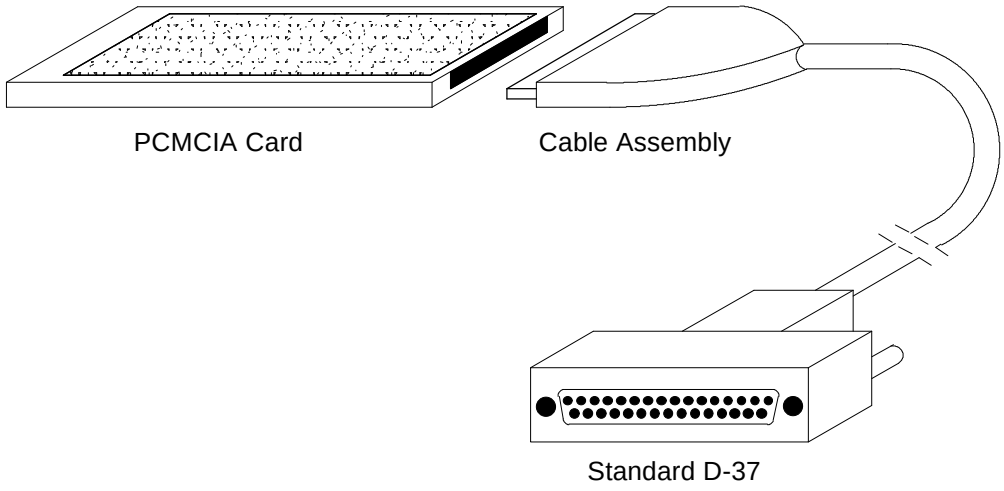


Figure 1. DA8P-12 Card and Cable Assembly

2. DOS / Windows 3.x Installation

Two configuration software programs are provided with the DA8P-12: a Client Driver, DA8P12CL.SYS, and a card Enabler, DA8P12EN.EXE. Either one of these programs may be used to configure the DA8P-12 hardware but only one may be used at a time. The table below highlights the differences between the Client Driver and the Enabler programs. The installation and usage of each of these programs is detailed in the sections that follow.

Client Driver	Enabler
DA8P12CL.SYS	DA8P12EN.EXE
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 DA8P-12 adapters upon insertion (Hot Swapping)	Does not support automatic configuration of DA8P-12 adapters upon insertion (Hot Swapping)
Requires PCMCIA Card and Socket Services software	Does not require PCMCIA Card and Socket Services software

On systems with Card and Socket Services installed, the Client Driver is the preferred method of installation. If you are unsure whether Card and Socket Services software is installed, install the DA8P-12 Client Driver as discussed in the following sections. When loaded, the Client Driver will display an error message if Card and Socket Services software is not detected.

2.1 DA8P-12 Client Driver for DOS

For systems using DOS and configured with PCMCIA Card and Socket Services software, the DA8P-12 includes a Client Driver, DA8P12CL.SYS, to enable and configure the adapter. PCMCIA Card and Socket Services software is not provided with the DA8P-12 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 DA8P-12 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 DA8P-12 Client Driver:

1. Copy the file DA8P12CL.SYS from the DA8P-12 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 in the CONFIG.SYS file where the Card and Socket Services software is installed.
4. **AFTER** the line installing the Card and Socket Services software, add the following line to the CONFIG.SYS file:

   DEVICE = *drive*:*path*\DA8P12CL.SYS *options*

   where *options* are the DA8P-12 Client Driver command line options discussed on the following pages.
5. Save the CONFIG.SYS file and exit the text editor.

NOTE: Since the DA8P-12 Client Driver supports "Hot Swapping", it is not necessary to have the DA8P-12 installed when booting the system. By inserting the DA8P-12 before booting, however, the Client Driver will report the adapter configuration during the boot process thereby verifying the changes made to the CONFIG.SYS.

6. Insert the DA8P-12 into one of the system's PCMCIA slots.

7. Reboot the system and note the message displayed when the DA8P-12 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 DA8P-12 Enabler program must be used to configure the adapter (see chapter 4). If the Client Driver reports the desired adapter configuration, the installation process is complete and the DA8P-12 may be removed and / or inserted from the system as desired. On each insertion into the PCMCIA socket, the DA8P-12 will automatically be re-configured to the specified settings.

2.1.2 Command Line Options

B*address* specifies the base I/O address of the DA8P-12 in hexadecimal. *address* must be in the range 100H - 3F8H and must reside on an even 8-byte boundary (*address* must end in 0 or 8). If the /B option is omitted, a base address will be assigned by Card and Socket Services.

I*irq* specifies the interrupt level (IRQ) of the DA8P-12 in hexadecimal. *irq* must be one of the following values: 3, 4, 5, 7, 9, A, B, C, E, F, or 0 if no IRQ is desired. If the /I option is omitted, an interrupt level will be assigned by Card and Socket Services.

S*socket* specifies the PCMCIA socket number to configure. *socket* must be in the range 0 - 15. The /S option is only required when multiple DA8P-12 adapters are to be installed in the same system.

2.1.3 Client Driver Examples

2.1.3.1 Example 1

```
DEVICE = C:\DA8P-12\DA8P12CL.SYS
```

In example 1, no command line arguments are specified. The Client Driver will configure any DA8P-12 inserted into any socket with a base address and IRQ assigned by Card and Socket Services.

2.1.3.2 Example 2

```
DEVICE = C:\DA8P-12\DA8P12CL.SYS (b330)
```

In example 2, a single command line argument is provided. The Client Driver will attempt to configure a DA8P-12 inserted into any socket at address 330H and an IRQ assigned by Card and Socket Services. If address 330H is unavailable, the DA8P-12 will not be configured.

2.1.3.3 Example 3

```
DEVICE = C:\DA8P-12\DA8P12CL.SYS (s0,b300,i5)
```

In example 3, a single command line argument is provided. The Client Driver will attempt to configure a DA8P-12 inserted into socket 0 with a base address of 300H and IRQ 5. If address 300H or IRQ 5 is unavailable, the DA8P-12 will not be configured. In addition, if a DA8P-12 is inserted into any other socket, it will not be configured.

2.1.3.4 Example 4

```
DEVICE = C:\DA8P-12\DA8P12CL.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 DA8P-12 inserted into any socket with a base address of 300H and IRQ 11. If address 300H or IRQ 11 is unavailable, the DA8P-12 will not be configured.

2.1.3.5 Example 5

```
DEVICE = C:\DA8P-12\DA8P12CL.SYS (b300,i5) (i10) ( )
```

In example 5, three command line arguments are provided. The Client Driver will first attempt to configure a DA8P-12inserted 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 DA8P-12 with a base address and an IRQ assigned by Card and Socket Services.

2.1.3.6 Example 6

```
DEVICE = C:\DA8P-12\DA8P12CL.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 DA8P-12 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.

2.1.3.7 Example 7

```
DEVICE = C:\DA8P-12\DA8P12CL.SYS (s0,b300,i5) (s1,b340,i10)
```

The type of configuration shown in example 7 may be desirable in systems where more than one DA8P-12 is to be installed. In this example, the Client Driver will attempt to configure a DA8P-12 inserted into socket 0 with a base address of 300H and IRQ 5. If the DA8P-12 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 DA8P-12'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 DA8P-12 will not be configured.

2.1.4 Examining the System Configuration

After the Client Driver is installed, the user may review the adapter configuration in two ways:

1. If the DA8P-12 is inserted in a PCMCIA slot before the system is booted, the adapter configuration is displayed when the Client Driver software is loaded. This configuration is guaranteed until the adapter is removed from the socket or the system is rebooted.
2. Most providers of Card and Socket Services software include a utility program to examine the configuration of any adapters installed in the PCMCIA sockets. These utilities can be executed anytime the system is powered-up and the adapter is installed. Consult the Card and Socket Services software documentation for information on this utility.

2.1.5 Where To Go From Here

The DA8P-12 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 5 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 DA8P-12.
2. For users who want to program the adapter with direct I/O transfers to the DA8P-12's register set, chapter 6 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 DA8P-12 directly should consult the DAQDRIVE software reference manual. DAQDRIVE provides a library of data acquisition subroutines for all Omega adapters and is included free of charge with the DA8P-12.
4. For TestPoint data acquisition software consult the documentation provided by the software manufacturer.

2.2 DA8P-12 Enabler for DOS

For systems that are not operating PCMCIA Card and Socket Services software, the DA8P-12 includes an Enabler program to enable and configure the adapter. This Enabler, DA8P12EN.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.

The DA8P-12 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 executingDA8P12EN.EXE. If more than one adapter is installed in a system, the Enabler must be executed separately for each adapter. Furthermore, DA8P12EN.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 re-configured with the Enabler after re-inserting it into a PCMCIA socket.

IMPORTANT:
The Enabler requires a region of high DOS memory when configuring aDA8P-12. This region is 1000H bytes long and by default begins at address D0000H (the default address may be changed using the Enabler's /M 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

S*socket* specifies the PCMCIA socket number to configure. *socket* must be in the range 0 - 7. Default: *socket* = 0.

B*address* specifies the base I/O address of the DA8P-12 in hexadecimal. *address* must be in the range 100H - 3F8H and must reside on an even 8-byte boundary (*address* must end in 0 or 8).

I*irq* specifies the interrupt level (IRQ) of the DA8P-12 in hexadecimal. *irq* must be one of the following values: 3, 4, 5, 7, 9, A, B, C, E, F, or 0 if no IRQ is desired.

L enables level sensitive interrupts (default is edge-triggered).

M*address* specifies the attribute memory window's base address in hexadecimal. Set *address*= D0 for a memory window at D0000, *address*= D8 for a memory windows at D8000, etc. *address* must be in the range C0 - F7. Default: *address*= D0.

R*socket* specifies the PCMCIA socket number to release. *socket* must be in the range 0 - 7. Default: *socket* = 0.

H displays the command line options.

2.2.2 Example Configurations

2.2.2.1 Example 1

DA8P12EN.EXE

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

2.2.2.2 Example 2

DA8P12EN.EXE (s0,b300,i5)

In example 2, the Enabler will configure the DA8P-12 in socket 0 with a base address of 300H and IRQ 5 using a configuration memory window at segment D000.

2.2.2.3 Example 3

DA8P12EN.EXE (i10,b340,s1)

In example 3, the Enabler will configure the DA8P-12 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.

2.2.2.4 Example 4

DA8P12EN.EXE (s0,b300,i3,wd8)

In example 4, the Enabler will configure the DA8P-12 in socket 0 with a base address of 300H and IRQ 3 using a configuration memory window at segment D800.

2.2.2.5 Example 5

DA8P12EN.EXE (s0,b300,i5,r)

In example 5, the Enabler will release the configuration used by the DA8P-12 in socket 0 using a configuration memory window at segment D000. The base address and IRQ parameters are ignored and may be omitted.

2.2.2.6 Example 6

(s1,r,wcc)

In example 5, the Enabler will release the configuration used by the DA8P-12 in socket 1 using a configuration memory window at segment CC00.

2.2.3 Where To Go From Here

The DA8P-12 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 5 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 DA8P-12.
2. For users who want to program the adapter with direct I/O transfers to the DA8P-12's register set, chapter 6 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 DA8P-12 directly should consult the DAQDRIVE software reference manual. DAQDRIVE provides a library of data acquisition subroutines for all Omega adapters and is included free of charge with the DA8P-12.
4. For TestPoint data acquisition software consult the documentation provided by the software manufacturer.

2.2.4 Common Problems

Memory Range Exclusion:
The Enabler requires a region of high DOS memory when configuring a DA8P-12. 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 DA8P-12's socket number to be specified on the command line and the DA8P-12 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 DA8P-12 Enabler, the lowest socket number in the system is designated socket 0.

Card and Socket Services Software:
In order to use the DA8P-12 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 Enabler may interfere with its operation and with the device(s) it controls. For systems configured with Card and Socket Services, the DA8P-12 Client Driver is the recommended method of configuration.

3. Windows 95 Installation

To allow easy configuration of the DA8P-12, an Windows 95 "**INF**" configuration file has been written for the hardware.

3.1 Installing a DA8P-12 Under Windows 95.

1. Insert the DA8P-12 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, “DA8P-12 Resource Settings".
3. The **New Hardware Found** window provides several options to configure the DA8P-12 card. Click the "**Driver from Disk**" option button. Click "OK" to continue.
4. An "**Install from Disk**" dialog box should appear. Insert the DA8P-12 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 DA8P-12 PC Card should now be configured. In the future, Windows 95 will automatically recognize and configure theDA8P-12.

3.2 DA8P-12 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 DA8P-12 belongs to this hardware class. The device name for the DA8P-12 is **Omega DA8P-12: PCMCIA Digital Input/Output Card.**
4. Open the **Properties** dialog for the DA8P-12 device, then click the **Resources** tab to view the Input/Output Range and Interrupt Request resource allocations (see Figure 3).
5. To access the DA8P-12 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 DA8P-12 belongs to this hardware class. The device name for the DA8P-12 is **Omega DA8P-12: PCMCIA Digital Input/Output Card.**
4. Open the **Properties** dialog for the DA8P-12 device, then click the **Resources** tab to view the Input/Output Range and Interrupt Request resource allocations.

C:\WINDOWS\da8p-12-om.bmp

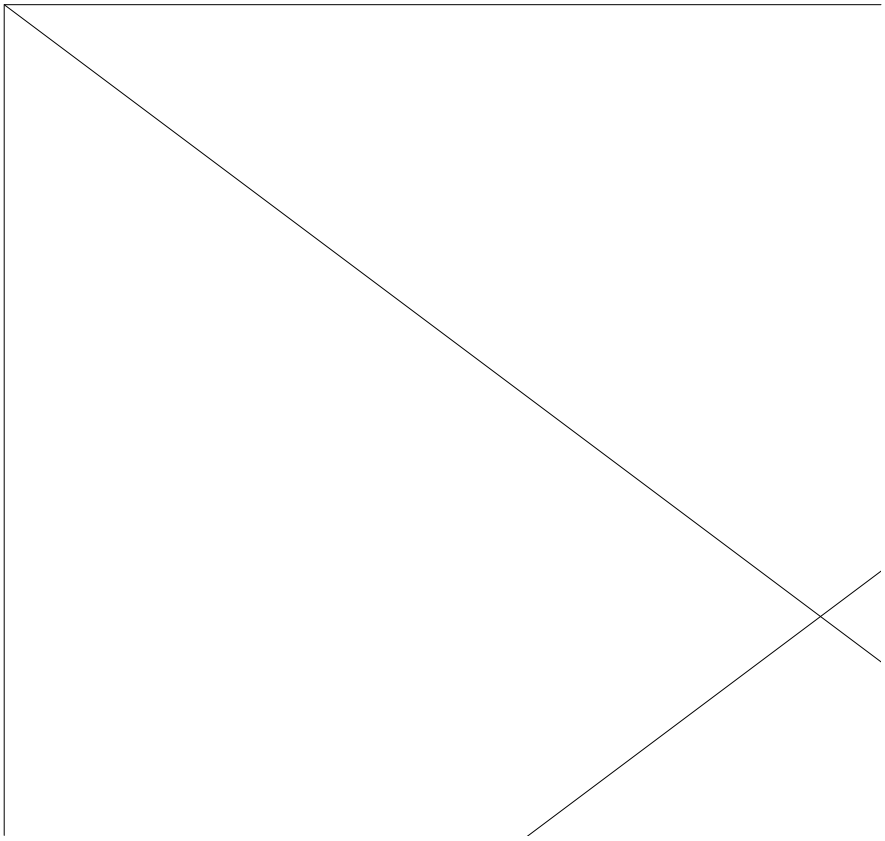


Figure 3. Windows 95 Resource Settings

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 DA8P-12. 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 DA8P-12'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 DA8P-12 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 DA8P-12.
2. For users who want to program the adapter with direct I/O transfers to the DA8P-12'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 DA8P-12 directly should consult the KPCMCIA-DAQWARE software reference manual. KPCMCIA-DAQWARE provides a library of data acquisition subroutines and is included free of charge with the DA8P-12.
4. For TestPoint data acquisition software consult the documentation provided by the software manufacturer.

4. Theory of Operation

4.1 Analog Outputs

The DA8P-12 is constructed with four dual-channel, 12-bit, serial load, D/A converters. The D/A converters may be written using 8 or 16-bit I/O instructions although 16-bit instructions are recommended. If 8 bit I/O transfers are used, the least significant byte (LSB) must be written before the most significant byte (MSB) as the contents of the data register are transferred to the appropriate D/A converter after the MSB is written.

Approximately 8µs is required to transfer the data into the D/A converter and no D/A converter may be written to until this transfer is complete. During non-simultaneous output operations, the analog output of the D/A converter begins changing as soon as all of the data bits are received and will settle to less than ½ LSB of the final value within 10µs under all conditions.

The analog outputs of the DA8P-12 are each rated for 1mA of load current. The total analog output current must remain less than 12mA or damage to the DA8P-12 may result.

WARNING:

The DA8P-12 may be permanently damaged if the total load on the analog outputs exceeds 12mA.

4.1.1 Data Formats

The bipolar DA8P-12B uses 2's complement digital codes ranging from -2048 to +2047. The unipolar DA8P-12U uses straight binary digital input codes ranging from 0 to 4095. Both adapters reset to 0 volts on power-up and after a software reset.

DA8P-12B		
Vout = (Value / 4096) * 10		
Value	Data bits DA11 ... DA0	Output
-2,048	1 0 0 0 0 0 0 0 0 0 0 0	-5.00000
-1	1 1 1 1 1 1 1 1 1 1 1 1	-0.00024
0	0 0 0 0 0 0 0 0 0 0 0 0	0.00000
1	0 0 0 0 0 0 0 0 0 0 0 1	0.00024
2,047	0 1 1 1 1 1 1 1 1 1 1 1	4.99756

DA8P-12U		
Vout = (Value / 4096) * 5		
Value	Data bits DA11 ... DA0	Output
0	0 0 0 0 0 0 0 0 0 0 0 0	0.00000
1	0 0 0 0 0 0 0 0 0 0 0 1	0.00122
2,047	0 1 1 1 1 1 1 1 1 1 1 1	2.49878
2,048	1 0 0 0 0 0 0 0 0 0 0 0	2.50000
4,095	1 1 1 1 1 1 1 1 1 1 1 1	4.99878

4.2 Simultaneous Analog Output

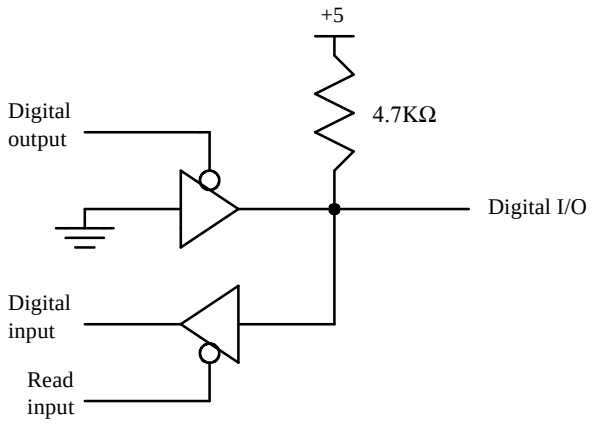
The DA8P-12 is capable of simultaneous output on 2 through 8 analog outputs. To perform simultaneous output, the associated simultaneous load bit(s) in the DA8P-12's simultaneous output register must be set to

logic 1 before any data is written to the D/A converters. As long as a channel's simultaneous load bit is set to logic 1, any data written to that D/A converter is held in an internal data buffer and **IS NOT** converted to an analog output voltage. After all of the D/A channels have been written, the analog outputs may be updated simultaneously by clearing the simultaneous load bits. The simultaneous load bits may be cleared under software control by writing logic 0s to the associated bit locations or under hardware control by generating a low-to-high transition on the external load control input (pin 30 on the I/O connector).

4.3 Digital Input / Output

The DA8P-12 is equipped with 8 bits of digital I/O which may be individually programmed as input or output. To use a digital I/O bit as an output, simply set the corresponding bit in the DA8P-12's digital I/O register to a logic 0 or logic 1 for the desired output value. To use a digital I/O bit as an input, the corresponding bit in the digital I/O register must be set to logic 1.

All of the digital I/O bits are reset to input mode on power-up and after a hardware reset. Each digital input is equipped with a 4.7KΩ pull-up resistor and will read a logic 1 if disconnected.



Each digital output is capable of sinking 4mA. The output source current is provided by a 4.7KΩ pull-up resistor on each output bit resulting in approximately 1mA of output current. If additional drive current is required, a 2KΩ or larger pull-up resistor may be connected external to the DA8P-12.

4.4 Timer

The DA8P-12 is equipped with an event timer to pace events under CPU or interrupt control. The timer circuit features an 8-bit software programmable timer which counts the output of a selectable pre-scaler. This pre-scaler allows the timer to count at a rate of 100KHz (100µs) or 1KHz (1ms) resulting in timer output rates ranging from 50KHz to 4Hz (200µs to 250ms).

Each time the timer expires, a timer event and optionally a timer interrupt is generated. This event and / or interrupt may be monitored by software to pace other operations. The timer output is also available on pin 27 of the I/O connector as a TTL level signal going high (logic 1) for 1 count each time the timer expires.

4.5 D/A Loading

The D/A Loading signal is a TTL level output available on pin 28 of the I/O connector. D/A Loading changes to a logic 0 state while data is being transferred to one of the eight D/A converters and returns to a logic 1 when the transfer is complete.

If the D/A converter being written IS NOT configured for simultaneous output mode, its analog output will begin to update to its new voltage level when the D/A Loading signal makes the low-to-high transition. The analog output will settle to within ½ LSB of the new voltage level within 10µs of this low-to-high transition.

If the D/A converter being written IS configured for simultaneous output mode, D/A Loading may be used to determine when the transfer is complete so that the External Load Control input does not occur until the data transfer is complete.

4.6 External Load Control

The External Load Control signal is a TTL compatible input on pin 30 of the I/O connector and may be used to control the loading of the D/A converters configured for simultaneous output mode. A low-to-high transition of the External Load Control input clears all of the simultaneous load bits in the DA8P-12's load control register causing any D/A converters configured for simultaneous output operation to update their analog output(s). The load event and the optional load event interrupt are also generated by the low-to-high transition of this input.

External Load Control is equipped with a 4.7KΩ pull-up resistor and may be left disconnected when not in use.

4.7 External Event

The DA8P-12 offers a TTL compatible external event input on pin 31 of the I/O connector which may be used to allow the CPU to monitor external hardware events. The external event and the optional external event interrupt are generated by a low-to-high transition of this input.

External Event is equipped with a 4.7KΩ pull-up resistor and may be left disconnected when not in use.

(This Page Left Intentionally Blank)

5. Register Descriptions

The DA8P-12 uses 8 consecutive I/O locations within the I/O address map of the system. The base address of the adapter is determined by the Client Driver or Enabler software programs as discussed in chapters 3 and 4. The next 8 I/O locations are used by the DA8P-12 for the following functions:

A2	A1	A0	I/O address	Register Description
0	0	0	base address + 0	D/A converter data - LSB
0	0	1	base address + 1	D/A converter data - MSB
0	1	0	base address + 2	Control Register
0	1	1	base address + 3	Simultaneous Output Register
1	0	0	base address + 4	Timer Register
1	0	1	base address + 5	Digital I/O
1	1	0	base address + 6	Interrupt Register
1	1	1	base address + 7	Reserved for future use

Each register of the DA8P-12 is discussed in detail in the following sections.

5.1 D/A Converter Data - LSB, MSB

The D/A converter data register, located at base address + 0 and base address + 1, is used to output data to one of the D/A converters. 16-bit I/O transfers should be used to access this register. If 8 bit I/O transfers must be used, the LSB register must be written before the MSB register as the contents of the data register are transferred to the D/A converter when the MSB register is written. After the MSB register is written, approximately 8µs is required to transfer the contents of the data register into the D/A converter. The data register must not be written to during this interval. The READY bit, located in the control register, may be used to determine when data may be safely written to the data register. The data register may be read at any time and returns the last value written.

The format of the data written to the data register is

	D7	D6	D5	D4	D3	D2	D1	D0
MSB	0	C2	C1	C0	DA11	DA10	DA9	DA8
LSB	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

where Cx selects the D/A converter

	C2	C1	C0
DAC 0	0	0	0
DAC 1	0	0	1
DAC 2	0	1	0
DAC 3	0	1	1
DAC 4	1	0	0
DAC 5	1	0	1
DAC 6	1	1	0
DAC 7	1	1	1

and DAx represents the data bits transferred to the D/A converter.

Decimal	Data bits DA11 ... DA0	DA8P-12B	DA8P-12U
-2,048	1 0 0 0 0 0 0 0 0 0 0 0	-5.00000	n/a
-1	1 1 1 1 1 1 1 1 1 1 1 1	-0.00244	n/a
0	0 0 0 0 0 0 0 0 0 0 0 0	0.00000	0.00000
1	0 0 0 0 0 0 0 0 0 0 0 1	0.00244	0.00122
2,047	0 1 1 1 1 1 1 1 1 1 1 1	4.99756	2.49878
2,048	1 0 0 0 0 0 0 0 0 0 0 0	n/a	2.50000
4,095	1 1 1 1 1 1 1 1 1 1 1 1	n/a	4.99878

5.2 Control Register

The control register, located at base address + 2, is used to control the DA8P-12's event timer and interrupt selections. On power-up, all bits in the control register are set to logic 0.

BIT	NAME	DESCRIPTION
7	RESET	During write operations, setting this bit to logic 1 generates a hardware reset to the D/A converters and the analog outputs are reset to 0 volts. RESET is self clearing.
	READY	During read operations, this bit indicates the status of the D/A converter control circuitry. When set to logic 1, the DA8P-12 is ready and data may be safely written to the D/A converter data register. When set to logic 0, the DA8P-12 is transferring the contents of the data register to one of the D/A converters. The data register must not be written while READY = 0.
6	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
5	TIMER_EVENT	During read operations, a logic 1 indicates the event timer has expired at least once since TIMER_EVENT was last reset. An interrupt may be generated from TIMER_EVENT by setting the TIMER_IRQ_EN bit in the DA8P-12's interrupt register. TIMER_EVENT is reset by writing a logic 1 to this bit location or by halting the event timer (setting TIMER_RUN = 0).
4	EXT_EVENT	During read operations, a logic 1 indicates at least one low-to-high transition of the external event input (pin 31 of the I/O connector) has occurred since EXT_EVENT was last reset. An interrupt may be generated from EXT_EVENT by setting the EXT_IRQ_EN bit in the DA8P-12's interrupt register. EXT_EVENT is reset by writing a logic 1 to this bit location.
3	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
2	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
1	TIMER_SOURCE	When set to logic 1, the event timer operates from a 100KHz clock source (100µs per count). When set to logic 0, the event timer operates from a 1KHz clock source (1ms per count).
0	TIMER_RUN	When set to logic 1, enables the operation of the event timer. When set to logic 0, the timer is disabled.

5.3 Simultaneous Output Register

The simultaneous output register, located at base address + 3, is used to control the simultaneous output capabilities of the DA8P-12. On power-up, all bits in the load register are set to logic 0.

BIT	NAME	DESCRIPTION
7	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
6	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
5	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
4	LOAD_EVENT	During read operations, a logic 1 indicates at least one low-to-high transition of the external load control input (pin 30 of the I/O connector) has occurred since LOAD_EVENT was last reset. An interrupt may be generated from LOAD_EVENT by setting the LOAD_IRQ_EN bit in the DA8P-12's interrupt control register. LOAD_EVENT is cleared by writing a logic 1 to this bit location.
3	LOAD_67	When set to logic 1, D/A converter channels 6 and 7 are configured for simultaneous output and any data written to their data registers is not converted to an analog output. Clearing LOAD_67 to logic 0 returns D/A channels 6 and 7 to the standard operating mode and causes the last value output to each of their data registers to be converted to an analog output. A low-to-high transition of the external load control input (pin 30 of the I/O connector) also clears LOAD_67 to logic 0. Additional information on simultaneous output operation is available in chapter 4.
2	LOAD_45	When set to logic 1, D/A converter channels 4 and 5 are configured for simultaneous output and any data written to their data registers is not converted to an analog output. Clearing LOAD_45 to logic 0 returns D/A channels 4 and 5 to the standard operating mode and causes the last value output to each of their data registers to be converted to an analog output. A low-to-high transition of the external load control input (pin 30 of the I/O connector) also clears LOAD_45 to logic 0. Additional information on simultaneous output operation is available in chapter 4.
1	LOAD_23	When set to logic 1, D/A converter channels 2 and 3 are configured for simultaneous output and any data written to their data registers is not converted to an analog output. Clearing LOAD_23 to logic 0 returns D/A channels 2 and 3 to the standard operating mode and causes the last value output to each of their data registers to be converted to an analog output. A low-to-high transition of the external load control input (pin 30 of the I/O connector) also clears LOAD_23 to logic 0. Additional information on simultaneous output operation is available in chapter 4.
0	LOAD_01	When set to logic 1, D/A converter channels 0 and 1 are configured for simultaneous output and any data written to their data registers is not converted to an analog output. Clearing LOAD_01 to logic 0 returns D/A channels 0 and 1 to the standard operating mode and causes the last value output to each of their data registers to be converted to an analog output. A low-to-high transition of the external load control input (pin 30 of the I/O connector) also clears LOAD_01 to logic 0. Additional information on simultaneous output operation is available in chapter 4.

5.4 Timer Register

The timer register, located at base address + 4, is used to set the rate of the on-board event timer. Each time the timer expires, a timer event and optionally a timer interrupt is generated (see the DA8P-12 control and interrupt registers). This event and / or interrupt may be monitored by software to pace other operations. The output of the timer is also available on pin 27 of the I/O connector.

The value written to the timer register determines the number of input clock cycles required before the timer expires and must be in the range $1 \leq$ timer register value ≤ 255. A read of the timer register returns the last value written. The timer's output rate (which is equal to the timer's event rate) may be calculated using the equations

$$\text{output rate (Hz)} = \frac{\text{input clock rate (Hz)}}{\text{register value + 1}}$$

- OR -

$$\text{output rate (sec)} = \text{input clock rate (sec)} * (\text{register value + 1})$$

where the timer's input clock rate is set to 100KHz (10µs per count) or 1KHz (1ms per count) using the TIMER_SOURCE bit located in the control register.

On power-up, the timer register is initialized to 0. A valid timer register value must be written to this register before the timer is enabled.

5.5 Digital I/O

The digital I/O register, located at base address + 5, directly controls the digital I/O signals on the output connector. To use a digital I/O bit as an output bit, simply write a logic 0 or a logic 1 to the associated location in the digital I/O register. To use a digital I/O bit as an input bit, a logic 1 must first be written to the associated bit in the digital I/O register. A read of the digital I/O register returns the current value of the digital inputs and the last value written to the digital outputs. On power-up, all digital I/O bits are set to input mode (logic 1).

5.6 Interrupt Register

The interrupt register, located at base address + 6, is used to enable and disable interrupts and to determine the source of an interrupt during interrupt processing.

BIT	NAME	DESCRIPTION
7	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
6	EXT_IRQ	During read operations, a logic 1 indicates an EXT_EVENT interrupt is pending. The external event interrupt must be disabled (EXT_IRQ_EN = 0) or the external event status must be reset (see EXT_EVENT in the DA8P-12 control register) before additional interrupts can be received. EXT_IRQ should be set to logic 0 during write accesses to this register.
5	LOAD_IRQ	During read operations, a logic 1 indicates a LOAD_EVENT interrupt is pending. The load event interrupt must be disabled (LOAD_IRQ_EN = 0) or the load event status must be reset (see LOAD_EVENT in the DA8P-12 simultaneous output register) before additional interrupts can be received. LOAD_IRQ should be set to logic 0 during write accesses to this register.
4	TIMER_IRQ	During read operations, a logic 1 indicates a TIMER_EVENT interrupt is pending. The timer event interrupt must be disabled (TIMER_IRQ_EN = 0) or the timer event status must be reset (see TIMER_EVENT in the DA8P-12 control register) before additional interrupts can be received. TIMER_IRQ should be set to logic 0 during write accesses to this register.
3	Reserved	This bit returns a logic 0 during read operations and must be set to logic 0 for write operations.
2	EXT_IRQ_EN	When set to logic 1, an interrupt is generated on each occurrence of EXT_EVENT (see EXT_EVENT in the DA8P-12 control register). When set to logic 0, external event interrupts are disabled.
1	LOAD_IRQ_EN	When set to logic 1, an interrupt is generated on each occurrence of LOAD_EVENT (see LOAD_EVENT in the DA8P-12 simultaneous output register). When set to logic 0, load event interrupts are disabled.
0	TIMER_IRQ_EN	When set to logic 1, an interrupt is generated on each occurrence of TIMER_EVENT (see TIMER_EVENT in the DA8P-12 control register). When set to logic 0, timer event interrupts are disabled.

6. I/O Connections

The DA8P-12 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).

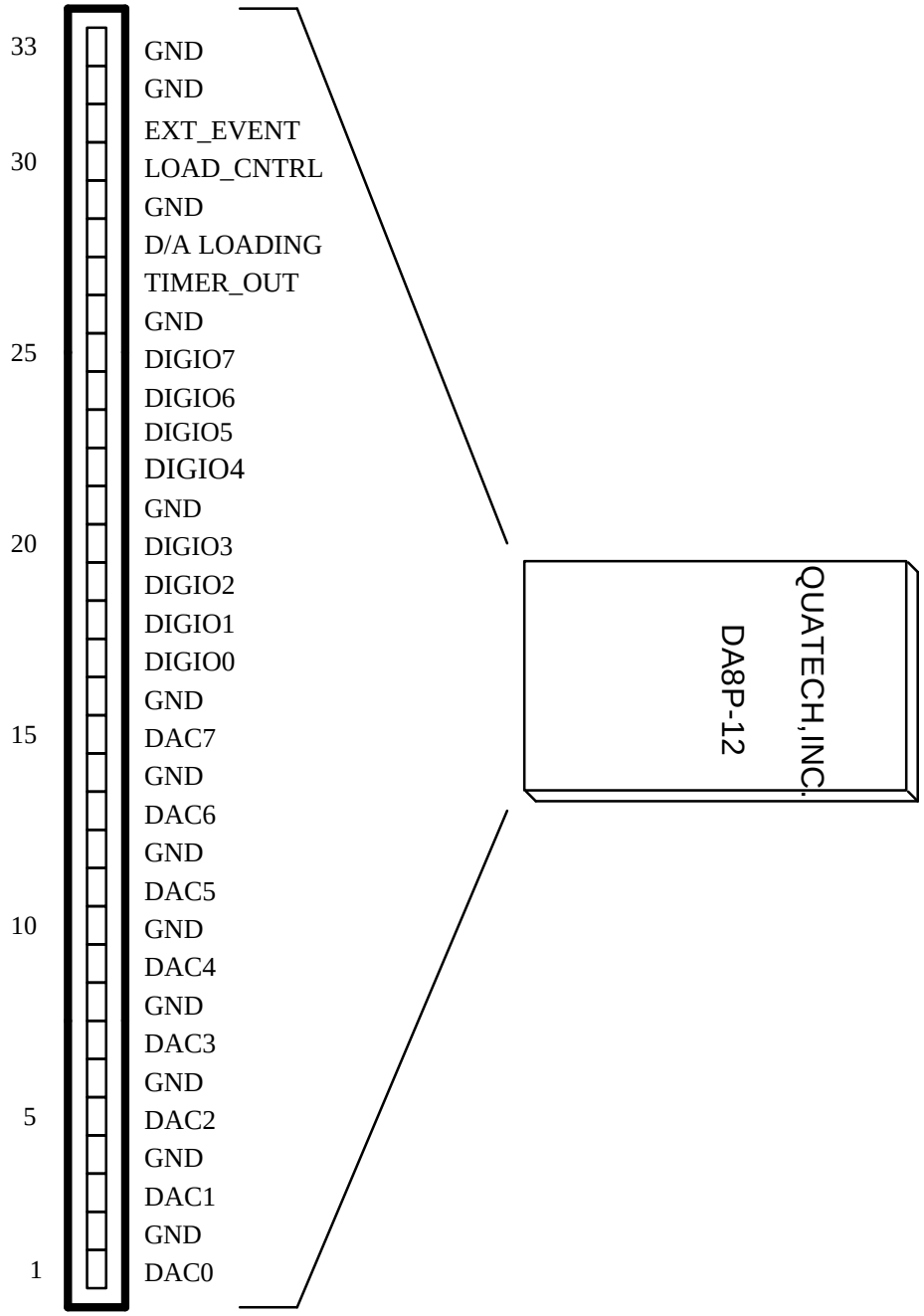


Figure 2. DA8P-12 output connector.

(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 DA8P-12's 33-pin 0.8mm I/O connector to a standard D-37 male connector. The first 31 connections on the DA8P-12 map directly to the first 31 pins of the D37 connector. Note that two of the DA8P-12's ground connections (pins 32 and 33) are not available when using the CP-IO37.

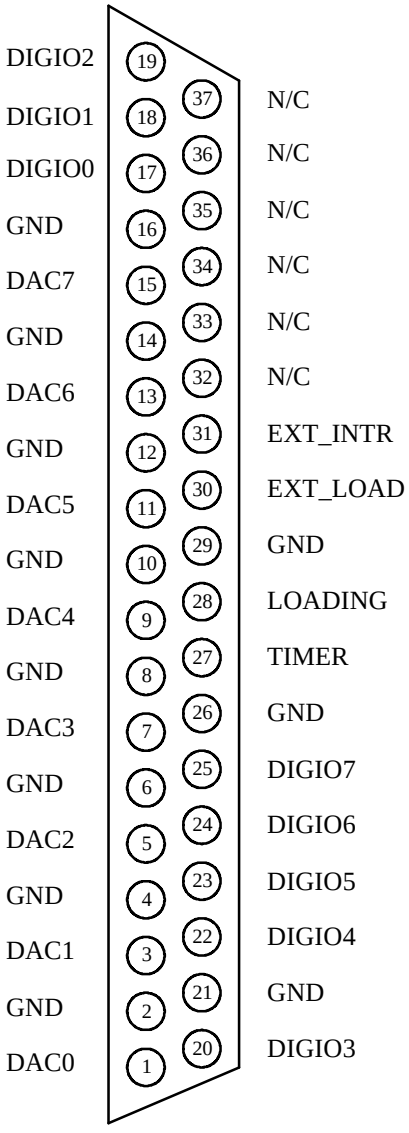


Figure 3. DA8P-12 output connections using the optional CP-IO37.

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 DA8P-12. 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 DA8P-12 are available on the CP-IO37 cable assembly, two of the DA8P-12's ground connections (pins 32 and 33) are not available when using the UIO-37.

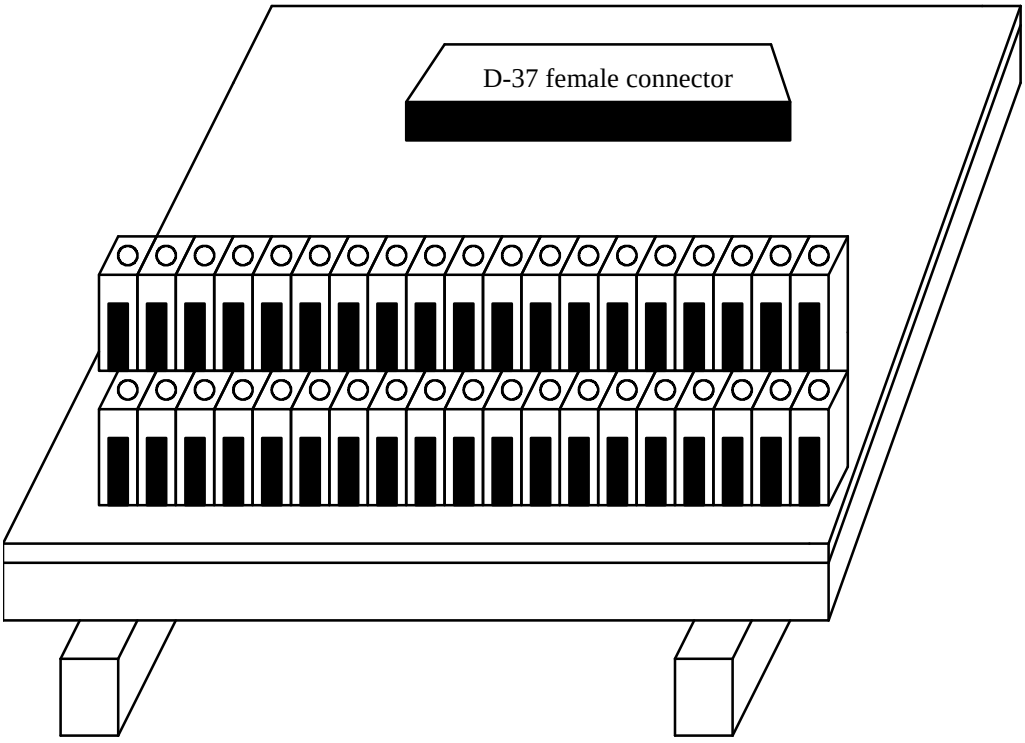


Figure 4. UIO-37 screw terminal adapter.

8. Specifications

Resolution	12 bits	
Relative Accuracy	± 1 LSB maximum	
Offset Error	± 5 LSB maximum	(DA8P-12U)
	± 6 LSB maximum	(DA8P-12B)
Full-Scale Error[1]	± 6 LSB maximum	
Settling Time to ±½ LSB		
Positive Full-Scale Change	10µs maximum	(3µs typical)
Negative Full-Scale Change	10µs maximum	(5µs typical)
Glitch-Impulse	30 nV-sec typical	
Digital Feedthrough	10nV-sec typical	
Digital Crosstalk	10nV-sec typical	
Output Load Current[2]	1mA per channel typical 12mA per adapter maximum	
Power Requirements		
No Load	120mA @ +5V	maximum
Full Rated Load	140mA @ +5V	maximum
Power-down Mode	20mA @ +5V	maximum

NOTES:

1. Full-Scale Error includes Offset Error
2. Total output load currents in excess of 12mA may result in damage to the DA8P-12.