

## **Copyright Notice**

This documentation and the software included with this product are copyrighted 2003 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the product described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, or for any infringements of the rights of third parties which may result from its use.

## **Acknowledgements**

PC-LabCard™ is a trademark of Advantech Co., Ltd.  
IBM and PC are trademarks of International Business Machines Corporation.  
MS-DOS, Windows, Microsoft Visual C++ and Visual BASIC are trademarks of Microsoft Corporation.  
Intel and Pentium are trademarks of Intel Corporation.  
Delphi and C++ Builder are trademarks of Borland Software Corporation.  
PICMG, CompactPCI and the PICMG, and CompactPCI logos are trademarks of the PCI Industrial Computers Manufacturers Group.  
All other product names or trademarks are properties of their respective owners.

## **CE notification**

The MIC-3716, developed by ADVANTECH CO., LTD., has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

## **On-line Technical Support**

For technical support and service, please visit our support website at:

**<http://www.advantech.com/support>**

Part No.: 2003371600

1st Edition

Printed in Taiwan

January 2004

*(This page is left blank for hard printing.)*

# Contents

<b>1. Introduction</b>	<b>2</b>
1.1 Product Description	3
1.2 What You Need to Get Started	6
1.3 Software Programming Choices	8
1.4 Optional Accessories	9
<b>2. Installation</b>	<b>12</b>
2.1 Unpacking	12
2.2 Software Installation	14
2.3 Hardware Installation	15
2.4 Device Setup & Configuration	20
2.5 Device Testing	24
<b>3. Signal Connections</b>	<b>32</b>
3.1 Overview	32
3.2 I/O Connector	32
3.3 Analog Input Connections	36
3.4 Analog Output Connections	41
3.5 Trigger Source Connections	42
3.6 Field Wiring Considerations	42
<b>4. Software Programming Overview</b>	<b>46</b>
4.1 Programming Choices	46
4.2 DLL Driver Programming Roadmap	47
<b>5. Calibration</b>	<b>52</b>
5.1 Overview	52
5.2 VR Assignment	53

5.3 Auto-Calibration .....	54
5.4 Manual Calibration.....	62
<b>Appendix A. Specifications .....</b>	<b>68</b>
<b>Appendix B. Block Diagrams .....</b>	<b>71</b>
<b>Appendix C. Register Structure and Format.....</b>	<b>73</b>
C.1 Overview.....	73
C.2 I/O Port Address Map .....	73
C.3 Channel Number and A/D Data — BASE+0 and BASE+1 .....	78
C.4 Software A/D Trigger — BASE+0.....	79
C.5 A/D Channel Range Setting — BASE+2.....	80
C.6 MUX Control — BASE+4 and BASE+5.....	82
C.7 Control Register — BASE+6 and BASE+7.....	86
C.8 Status Register — BASE+6 and BASE+7 .....	89
C.9 Clear Interrupt and FIFO — BASE+8 and BASE+9.....	90
C.10 D/A Output Channel 0 — BASE+10 and BASE+11 .....	90
C.11 D/A Output Channel 0 — BASE+10 and BASE+11 .....	91
C.12 D/A Output Channel 1 — BASE+12 and BASE+13.....	92
C.13 D/A Output Channel 1 — BASE+12 and BASE+13.....	93
C.14 D/A Reference Control —BASE+14 .....	94
C.15 Digital I/O Registers — BASE+16 and BASE+17.....	95
C.16 Calibration Registers — BASE+18 and BASE+19 .....	96
C.17 Board ID Registers — BASE+20.....	98
C.18 Programmable Timer/Counter Registers BASE+24, BASE+26, BASE+28 and BASE+30.....	98
<b>Appendix D. 82C54 Counter Chip Function .....</b>	<b>99</b>
D.1 The Intel 82C54 .....	99

D.2 Counter Read/Write and Control Registers.....	101
D.3 Counter Operating Modes.....	105
D.4 Counter Operations.....	108
<b>Appendix E. Manual Calibration .....</b>	<b>110</b>
E.1 A/D Calibration .....	111
E.2 D/A Calibration .....	114

*(This page is left blank for hard printing.)*

# Tables

Table 3-1: I/O Connector Signal Description .....	35
Table C-1: MIC-3716 register format (Part 1) .....	74
Table C-1: MIC-3716 register format (Part 2) .....	75
Table C-1: MIC-3716 register format (Part 3) .....	76
Table C-1: MIC-3716 register format (Part 4) .....	77
Table C-2: MIC-3716 Register for A/D data .....	78
Table C-3: Register for A/D channel range setting .....	80
Table C-4: Gain codes for MIC-3716 .....	81
Table C-5: Register for multiplexer control .....	82
Table C-6: Control Register .....	86
Table C-7: MIC-3716 Register for channel number and A/D data .....	87
Table C-8: Status Register .....	89
Table C-9: Register to clear interrupt and FIFO .....	90
Table C-10: Register for load D/A channel 0 data .....	90
Table C-11: Register for D/A channel 0 data .....	91
Table C-12: Register for load D/A channel 1 data .....	92
Table C-13: Register for D/A channel 1 data .....	93
Table C-14: Register for D/A reference control .....	94
Table C-15: Register for digital input .....	95
Table C-16: Register for digital output .....	95
Table C-17: Calibration Command and Data Register .....	96
Table C-18: Calibration Command and Data Register .....	97
Table C-19: Register for Board ID .....	98
Table E-1: A/D binary code table .....	113

Table E-2: D/A binary code table.....**117**

# Figures

Fig. 1-1: Installation Flow Chart	7
Fig. 2-1: The Setup Screen of Advantech Automation Software	14
Fig. 2-2: Different options for Driver Setup	15
Fig. 2-3: The device name listed on the Device Manager	19
Fig. 2-4: The Device Manager dialog box	21
Fig. 2-5: The Device Setting dialog box	22
Fig. 2-6: The Device Name appearing on the list of devices box	23
Fig. 2-7: The Advantech Device Test dialog box	24
Fig. 2-8: Analog Input tab on the Device Test dialog box	25
Fig. 2-9: Analog Output tab on the Device Test dialog box	26
Fig. 2-10: Digital Input tab on the Device Test dialog box	27
Fig. 2-11: Digital Output tab on the Device Test dialog box	28
Fig. 2-12: Counter tab on the Device Test dialog box	29
Fig. 3-1: I/O connector pin assignments for the MIC-3716	34
Fig. 3-2: Single-ended input channel connection	37
Fig. 3-3: Differential input channel connection - ground reference signal source	38
Fig. 3-4: Differential input channel connection - floating signal source	39
Fig. 3-5: Analog output connections	41
Fig. 5-1: MIC-3716 VR assignment	54
Fig. 5-2: Selecting the device you want to calibrate	55
Fig. 5-3: Warning message before start calibration	56
Fig. 5-4: Auto A/D Calibration Dialog Box	57
Fig. 5-5: A/D Calibration Procedure 1	57

Fig. 5-6: A/D Calibration Procedure 2.....	<b>58</b>
Fig. 5-7: A/D Calibration Procedure 3.....	<b>58</b>
Fig. 5-8: A/D Calibration is finished.....	<b>59</b>
Fig. 5-9: Range Selection in D/A Calibration .....	<b>60</b>
Fig. 5-10: Calibrating D/A Channel 0.....	<b>60</b>
Fig. 5-11: Calibrating D/A Channel 1 .....	<b>61</b>
Fig. 5-12: D/A Calibration is finished .....	<b>61</b>
Fig. 5-13: Selecting Input Range in Manual A/D Calibration panel.....	<b>63</b>
Fig. 5-14: Adjusting registers .....	<b>63</b>
Fig. 5-15 & Fig. 5-16: Selecting D/A Range and Choosing Output Voltage.....	<b>65</b>
Fig. 5-17: Adjusting registers .....	<b>66</b>

**CHAPTER**

**1**

## **Introduction**

# 1. Introduction

---

Thank you for buying the Advantech MIC-3716 CompactPCI (cPCI) card. The Advantech MIC-3716 is a powerful data acquisition (DAS) card which complies with PICMG 2.0 R2.1 CompactPCI specifications. It features a unique circuit design and complete functions for data acquisition and control, including A/D conversion, D/A conversion, digital input, digital output, and counter/timer. MIC-3716 provides specific functions for different user requirements.

The following sections of this chapter will provide further information about features of the multifunction cards, a quick starting guide for installation, as well as some brief information on software and accessories for the MIC-3716 card.

## 1.1 Product Description

The Advantech MIC-3716 provides users with the most requested measurement and control functions as below:

- ◆ Compact PCI bus specification 2.1 compliant
- ◆ PCI-bus mastering for data transfer
- ◆ 16-channel Single-Ended or 8 differential analog input
- ◆ 16-bit A/D conversion with up to 250 KS/s sampling rate
- ◆ Programmable gain for each input channel
- ◆ 1 K on board samples FIFO buffer
- ◆ 2-channel analog output
- ◆ 16-channel digital input
- ◆ 16-channel digital output
- ◆ Programmable counter/timer
- ◆ Automatic channel/gain scanning
- ◆ I/O address automatically assigned by PCI plug-and-play
- ◆ Board ID

The Advantech MIC-3716 offers the following main features:

### Plug-and-Play Function

The Advantech MIC-3716 is a Plug-and-Play device that fully complies with the PCI Specification Rev 2.2. During card installation, all bus-related configurations such as base I/O address and interrupts are conveniently taken care of by the Plug-and-Play function. You have virtually no need to set any jumpers or DIP switches.

## **Flexible Input Type and Range Settings**

The MIC-3716 features an automatic channel/gain scanning circuit. This circuit design controls multiplexer switching during sampling. Users can set different gain values for each channel according to their needs for the corresponding range of input voltage. The gain value settings thus selected is stored in the SRAM. This flexible design enables multi-channel and high-speed sampling for high-performance data acquisition.

## **On-board FIFO (First-In-First-Out) Memory**

The MIC-3716 provides an onboard FIFO memory buffer that can store 1K samples. Users can either enable or disable the interrupt request feature of the FIFO buffer. While the interrupt request for FIFO is enabled, users are allowed to specify whether an interrupt request will be sent with each sampling action or only when the FIFO buffer is half saturated. This useful feature enables a continuous high-speed data transfer with a more predictable performance on operating systems.

## **16 Digital Inputs and 16 Digital Outputs**

The MIC-3716 provides 16 digital input channels and 16 digital output channels. This leaves you with great flexibility to design and customize your applications according to your specific needs.

## **On-board Programmable Counter**

The MIC-3716 is equipped with a programmable counter, which can

serve as a pacer trigger for A/D conversions. The counter chip is an 82C54 or its equivalent, which incorporates three 16-bit counters on a 10 MHz clock. One of the three counters is used as an event counter for input channels or pulse generation. The other two are cascaded into a 32-bit timer for pacer triggering.

## Short Circuit Protection

The MIC-3716 is equipped with short circuit protection device (polyswitch) on the  $+12V_{DC}/+5V_{DC}$  power supply pins. If any of the power supply pins are shorted to ground (i.e. short circuit occurs), the protection device will shut off the current output automatically. After the short circuit has been released for about two minutes, the power supply pins will return to output current.

## Board ID

The MIC-3716 has a built-in DIP switch that helps define each card's ID when multiple cards have been installed on the same CompactPCI system. The board ID function is very useful when users build their system with multiple MIC-3716 cards. With the correct Board ID settings, you can easily identify and access each card during hardware configuration and software programming.

---

### Note:

1. Pace trigger determines how fast A/D conversion will be done in pacer trigger mode.
  2. For detailed specifications of the MIC-3716, please refer to *Appendix A, Specifications*.
-

## 1.2 What You Need to Get Started

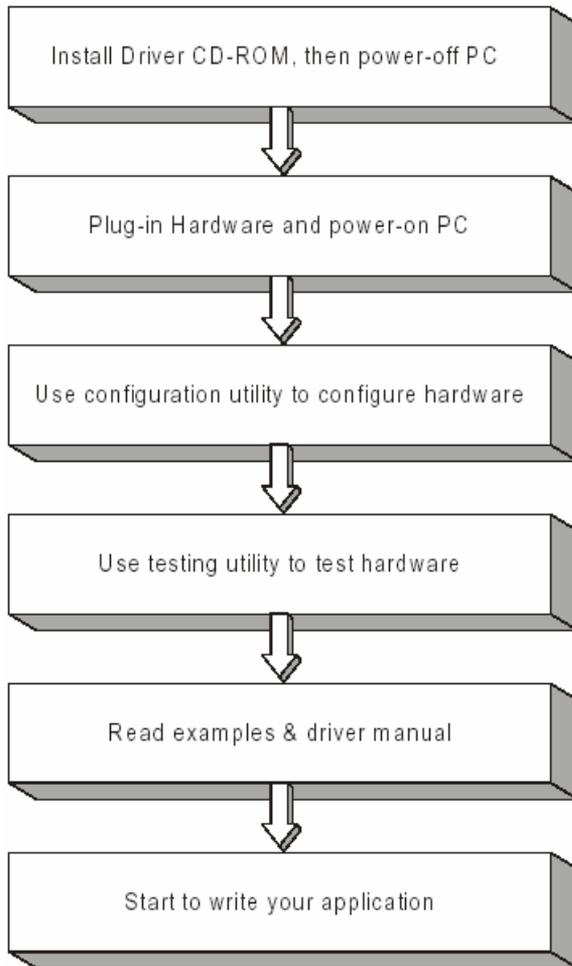
Before you install your MIC-3716 card, please make sure you have the following necessary components:

- ◆ **MIC-3716 Multifunction card**
- ◆ **MIC-3716 User's Manual**
- ◆ **Driver software**      Advantech DLL drivers (included in the companion CD-ROM)
- ◆ **Wiring cable**          PCL-10168
- ◆ **Wiring board**          PCLD-8710, ADAM-3968
- ◆ **Computer**              Personal computer or workstation with a CompactPCI-bus slot (running *Windows 95/98/NT/2000/XP*)

Some other optional components are also available for enhanced operation:

- ◆ **Application software** ActiveDAQ, GeniDAQ or other third-party software packages

After you get the necessary components and maybe some of the accessories for enhanced operation of your Multifunction card, you can begin the Installation procedure. *Fig. 1-1* on the next page provides a concise flow chart to give you a broad picture of the software and hardware installation procedures:



***Fig. 1-1: Installation Flow Chart***

### 1.3 Software Programming Choices

Advantech offers a rich set of DLL drivers, third-party driver support and application software to help fully exploit the functions of your MIC-3716 card:

- ◆ DLL driver (on the companion CD-ROM)
- ◆ LabVIEW driver
- ◆ Advantech ActiveDAQ
- ◆ Advantech GeniDAQ

For more information on software, please refer to *Chapter 4, Software Programming Overview*.

Users who intend to program directly at the registers of the multifunction card can have register-level programming as an option. Since register-level programming is often difficult and laborious, it is usually recommended only for experienced programmers. For more information, please refer to *Appendix C, Register Structure and Format*.

## 1.4 Optional Accessories

Advantech offers a complete set of accessory products to support the MIC-3716 cards. These accessories include:

### Wiring Cable

- **PCL-10168** The PCL-10168 shielded cable is specially designed for MIC-3716 cards to provide high resistance to noise. To achieve a better signal quality, the signal wires are twisted in such a way as to form a “twisted-pair cable”, reducing cross-talk and noise from other signal sources. Furthermore, its analog and digital lines are separately sheathed and shielded to neutralize EMI/EMC problems.

### Wiring Boards

- **ADAM-3968** The ADAM-3968 is a 68-pin SCSI wiring terminal module for DIN-rail mounting. This terminal module can be readily connected to Advantech PC-LabCard products and allow easy yet reliable access to individual pin connections for the MIC-3716 card.
  
- **PCLD-8710** The PCLD-8710 is a DIN-rail mounting screw-terminal board to be used with any of the PC-LabCard products which have 68-pin SCSI

connectors. The PCLD-8710 features the following functions:

- ◆ Two additional 20-pin flat-cable connectors for digital input and output
- ◆ Reserved space on the board to meet future needs for signal-conditioning circuits (e.g. low-pass filter, voltage attenuator and current shunt)
- ◆ Industrial-grade screw-clamp terminal blocks for heavy-duty and reliable connections.

**CHAPTER** **2**

**Installation and  
Configuration**

## 2. Installation

---

This chapter gives users a package item checklist, proper instructions about unpacking and step-by-step procedures for both driver and card installation.

### 2.1 Unpacking

After receiving your MIC-3716 package, please inspect its contents first. The package should contain the following items:

- MIC-3716 card
- Companion CD-ROM (DLL driver included)
- User's Manual

The MIC-3716 card harbors certain electronic components vulnerable to electrostatic discharge (ESD). ESD could easily damage the integrated circuits and certain components if preventive measures are not carefully paid attention to. **Before removing the card from the antistatic plastic bag, you should take following precautions to ward off possible ESD damage:**

- ◆ Touch the metal part of your computer chassis with your hand to discharge static electricity accumulated on your body. Or one can also use a grounding strap.
- ◆ Make contact between the antistatic bag and a metal part of the computer chassis before opening the bag.
- ◆ Hold the card by the metal bracket only when removing it from its bag.

**After taking out the card, you should first:**

- ◆ Inspect the card for any possible signs of external damage (loose or damaged components, etc.). If the card is visibly damaged, please notify our service department or our local sales representative immediately. Avoid installing a damaged card in your system.

**Also pay extra caution to the following aspects to ensure proper installation:**

- ◆ Avoid physical contact with materials that could hold static electricity such as plastic, vinyl and styrofoam™.
- ◆ Whenever you handle the card, hold it only by its edges. **DO NOT TOUCH** the exposed metal pins of the connector or the electronic components.

---

**Note:**

Keep the antistatic bag for future use. You might need the original bag to store the card if you have to remove the card from PC or transport it elsewhere.

---

## 2.2 Software Installation

We recommend you install the driver before you plug the MIC-3716 card into your system, since this will guarantee a smooth installation process.

The 32-bit DLL driver Setup program for the MIC-3716 card is included on the companion CD-ROM that is shipped with your DAS card package. Please follow the steps below to install the driver software:

**Step 1:** Insert the companion CD-ROM into your CD-ROM drive.

**Step 2:** The *Setup* program will be launched automatically if you have the *Autorun* function enabled on your system. When the *Setup* program is launched, you'll see the following setup screen.



**Fig. 2-1: The Setup Screen of Advantech Automation Software**

---

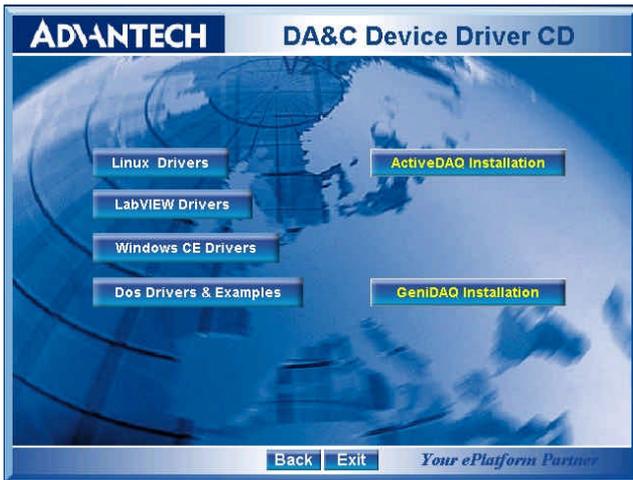
**Note:**

If the autoplay function is not enabled on your computer, use Windows Explorer or Windows *Run* command to execute SETUP.EXE on the companion CD-ROM.

---

**Step 3:** Select the *Installation* option, then the *Individual Drivers* option.

**Step 4:** Select the specific device then just follow the installation instructions step by step to complete your device driver setup.



**Fig. 2-2: Different options for Driver Setup**

For further information on driver-related issues, an online version of the *DLL Drivers Manual* is available by accessing the following path:  
*Start/ Programs/ Advantech Automation/ Device Manager/ Device Driver's Manual*

## 2.3 Hardware Installation

---

**Note:**

Make sure you have installed the driver before you install the card (please refer to *2.2 Software Installation*)

---

After the DLL driver installation is completed, you can install the MIC-3716 card in a CompactPCI slot on your computer.

It is recommended that you refer to your computer's user manual or related documentations if you have any problems. Please follow the steps below to install the card in your system:

**To install a card**

**Step 1:** Remove the cover of an unused cPCI slot on your computer.

**Step 2:** Hold the card vertically. Be sure that the card is pointing in the correct direction. The components of the card should be pointing to the right-hand side and the black handle of the card should be pointing to lower edge of the chassis.

**Step 3:** Holding the lower handle, pull the handle down to unlock it.

**Step 4:** Insert the MIC-3716 card into the cPCI chassis carefully by sliding the lower edge of the card into the card guide.

**Step 5:** Push the card into the slot gently by sliding the card along the card guide until J1 meets the long needle on the backplane.

---

**Note:**

If your card is correctly positioned and has been slid all the way into the chassis, the handle should match the rectangular holes. If not, remove the card from the card guide and repeat the steps from **step 2**. Do not try to install your card by forcing it into the chassis.

---

**Step 6:** Push the card into the right place, and secure it by pushing the handle to lock it into place.

---

**Note**

If the system power is on, the **Blue LED** on the front panel details the installation status of the card while the system is active. In **Step 5**, when J1 meets the long needle on the backplane, the **Blue LED** will be lit. After **Step 6**, the system can configure the card automatically, and the **Blue LED** turns off when the system finishes the device configuration.

---

**To remove a card**

**Step 1:** Push the handle down to unlock the card. The cPCI system will automatically start to uninstall the card configuration.

**Step 2:** Once the system has finished uninstalling the device configuration, the **Blue LED** on the front panel will turn on. You can now slide the card out.

---

**Note :**

1. Advantech MIC-3761's "Hot-Swap" function complies with the cPCI Hot Swap Specification PICMG 2.1 R2.0.
  2. Because of the "Hot-Swap", the above steps detail the card removal process while the system is on. If the system power is off, please follow **Step1** and **Step2** and disregard the status of the **Blue LED**.
-

---

**Note:**

If you installed the card without installing the DLL driver first, *Windows 95/98* will not recognize your card and label it as an “unknown device”. After rebooting, you will be prompted to provide the necessary driver. You should ignore the prompting messages (just click the **Cancel** button) and set up the driver according to the steps described in *2.2 Software Installation*.

---

After the MIC-3716 card is installed, you can verify whether it is properly installed on your system with the *Device Manager*:

1. Access the *Device Manager* through:  
*Start/ Control Panel/ System/ Device Manager*
2. The device name of the MIC-3716 should be listed on the *Device Manager*.



*Fig. 2-3: The device name listed on the Device Manager*

---

**Note:**

If your card is properly installed, you should see the *device name* of your card listed on the *Device Manager* tab. **If you do see your device name listed on it but marked with an exclamation sign “!”, it means your card has not been correctly installed.** In this case, remove the card device from the *Device Manager* by selecting its device name and press the **Remove** button. Then go through the driver installation process again.

---

After your card is properly installed on your system, you can now configure your device using the *Device Manager* program that has itself already been installed on your system during driver setup. A complete device installation procedure should include device setup, configuration and testing. The following sections will guide you through the *setup, configuration* and *testing* of your device.

## 2.4 Device Setup & Configuration

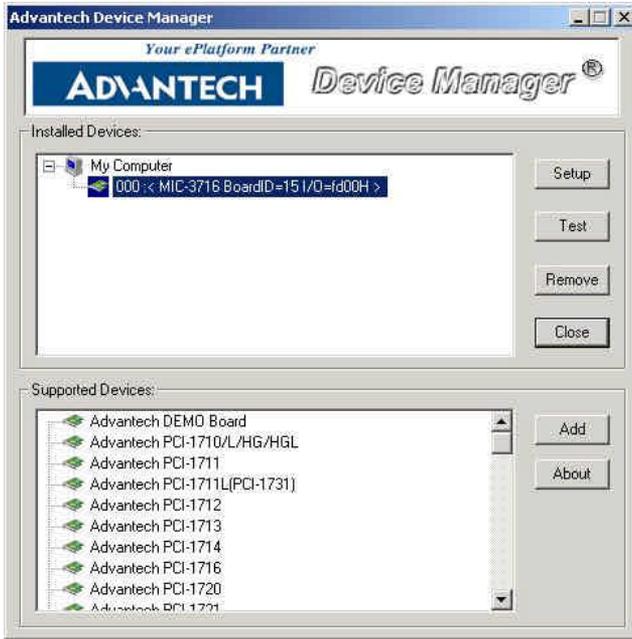
The *Device Manager* program is a utility that allows you to set up, configure and test your device, and later stores your settings on the system registry. These settings will be used when you call the APIs of *Advantech Device Drivers*.

### Setting Up the Device

**Step 1:** To install the I/O device for your card, you must first run the *Advantech Device Manager* program by accessing:  
*Start/ Programs/ Advantech Automation/ Device Manager/ Advantech Device Manager.*

**Step 2:** The *Advantech Device Manager* will search for the device(s) already installed on your system (if any) automatically.

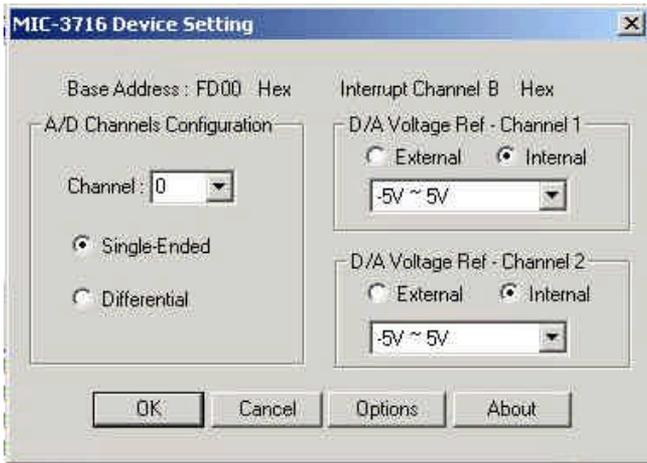
**Step 3:** The *Installed Devices* box will list the device(s) found on your system, such as the one in *Fig. 2-4*.



***Fig. 2-4: The Device Manager dialog box***

## Configuring the Device

**Step 4:** In the *Device Setting* dialog box (Fig. 2-5), you can configure the voltage source either as External or Internal, and specify the voltage output range for the two D/A channels.



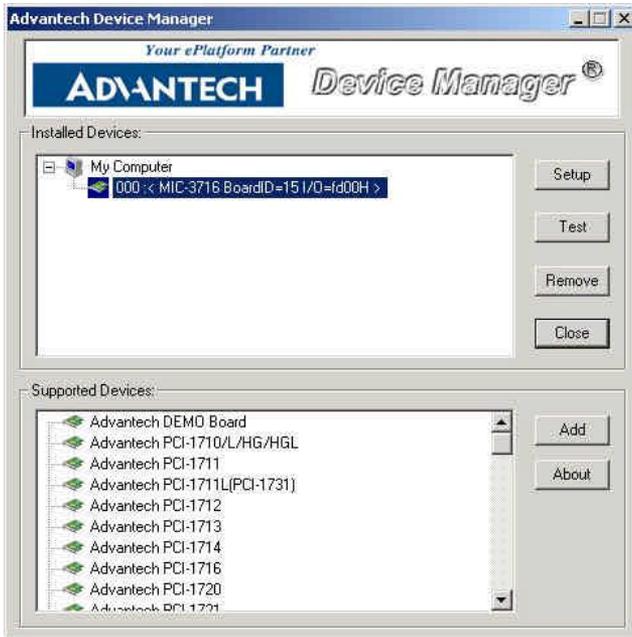
**Fig. 2-5: The Device Setting dialog box**

---

### Note:

- ◆ You can configure the source of D/A reference voltage either as *Internal* or *External*, and select the output voltage range. When selecting voltage source as **Internal**, you have two options for the output voltage range : **0 ~ 5 V** and **0 ~ 10 V**.
  - ◆ When selected as **External**, the output voltage range is determined by the external reference voltage in the following way :
  - ◆ **By inputting an external reference voltage: -xV , where |x| <= 10, you will get a output voltage range: 0 to xV.**
-

**Step 5:** After you have finished configuring the device, click **OK** and the device name will appear in the *Installed Devices* box as shown in *Fig. 2-6*.



**Fig. 2-6:** *The Device Name appears in the list of devices dialog box*

---

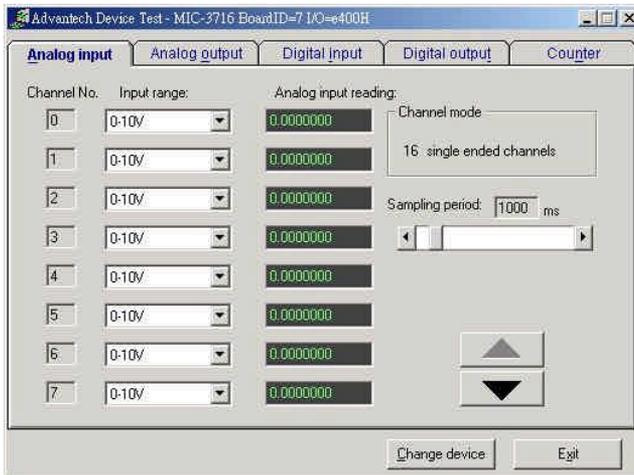
**Note:**

As you can see from the above figure, the device name is: **“000: <MIC-3716 BoardID=15 I/O=fd00H >”**. Each device name begins with a device number (“000”), which is specifically assigned to each card. The device number is passed to the driver to specify which device you wish to control.

---

## 2.5 Device Testing

Following through the *Setup* and *Configuration* procedures to the last step described in the previous section, you can now proceed to test the device by clicking the **Test** Button on the *Device Manager* dialog box. An *Advantech Device Test* dialog box will appear accordingly (Fig. 2-7).



**Fig. 2-7: The Advantech Device Test dialog box**

In the *Advantech Device Test* dialog box, you can test various functions of MIC-3716 from the *Analog input*, *Digital input*, *Digital output*, *Analog output* or *Counter* tabs.

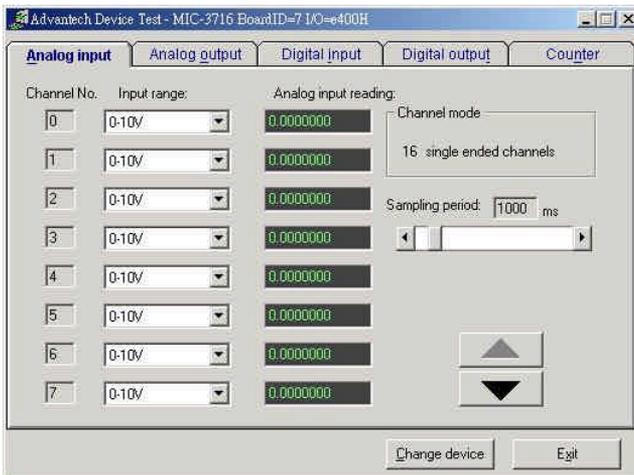
---

**Note:**

- You can access the *Advantech Device Test* dialog box either by the previous procedure for the *Advantech Device Manager* or simply by accessing *Start/Programs/ Advantech Automation/ Device Manager/ Advantech Device Manager*.
  - All the functions are performed by the software polling method. For high speed data acquirement or output, they have to use corresponding VC examples like ADINT, ADDMA or ADBMDMA.
- 

## Testing the Analog Input Functions

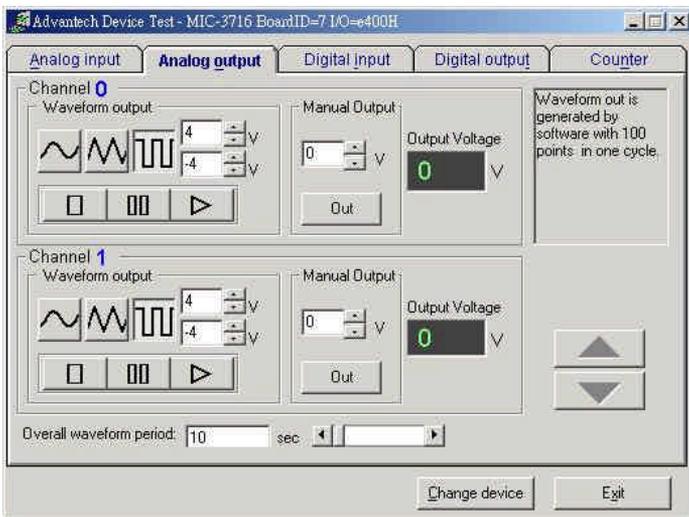
Click the *Analog Input* tab to bring it up to the front of the screen. Select the input range for each channel in the *Input range* drop-down boxes. Configure the sampling rate by adjusting the sampling period with the scroll bar. Switch between the channels by using the up/down arrow.



**Fig. 2-8: Analog Input tab in the Device Test dialog box**

## Testing the Analog Output Function

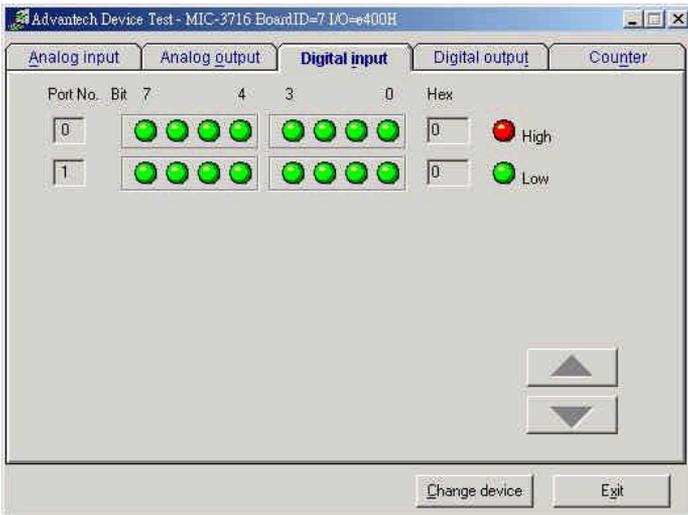
Click the *Analog Output* tab to bring it up to the foreground. The *Analog Output* tab allows you to output quasi-sine, triangle, or square waveforms generated by the software automatically, or output single values manually. You can also configure the waveform frequency and output voltage range.



**Fig. 2-9:** *Analog Output tab on the Device Test dialog box*

## Testing the Digital Input Functions

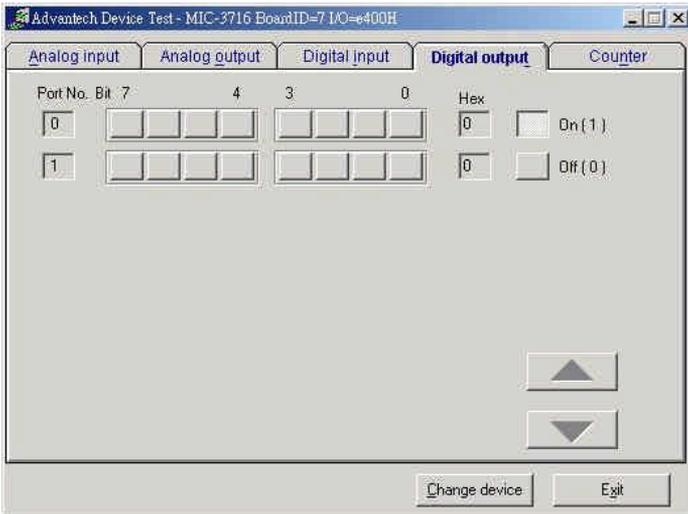
Click the *Digital Input* tab to show the *Digital Input* test panel as seen below in figure 2-10. Through the color of the lamps, you can discern whether the status of each digital input channel is either high or low. (Red = high, Green = low)



**Fig. 2-10: Digital Input tab on the Device Test dialog box**

## Testing the Digital Output Functions

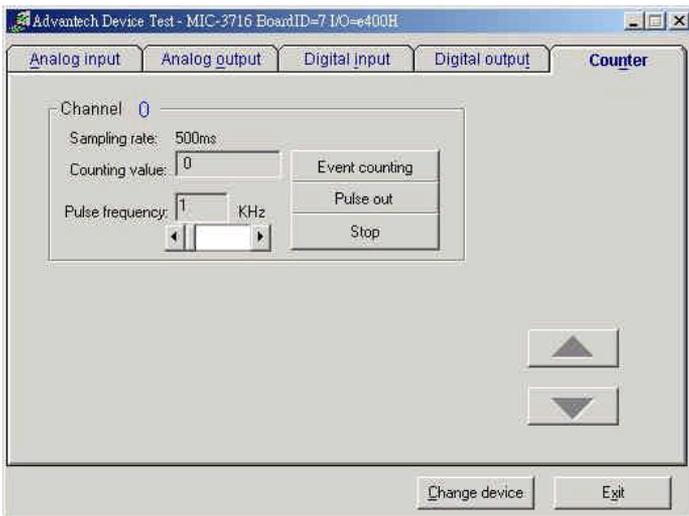
Click the *Digital Output* tab to bring up the *Digital Output* test panel such as the one seen on the next page. By pressing the buttons on each tab, you can set each digital output channel as *high* or *low* for the corresponding port.



**Fig. 2-11: Digital Output tab on the Device Test dialog box**

## Testing the Counter Function

Click the *Counter* Tab to bring its test panel forth. The counter channel (*Channel 0*) offers two options: Event counting and Pulse out. If you select Event counting, you need to first connect your clock source to pin CNT0\_CLK. The counter will start counting after the pin CNT0\_GATE is triggered. If you select Pulse Out, the clock source will be output to pin CNT0\_OUT. You can configure the *Pulse Frequency* with the scroll bar.



**Fig. 2-12: Counter tab on the Device Test dialog box**

Only after your card device is properly set up, configured and tested, can the device installation procedure be counted as complete. After the device installation procedure is completed, you can safely proceed to the next chapter, *Signal Connections*.

*(This page is left blank for hard printing.)*

CHAPTER

3

## Signal Connections

## 3. Signal Connections

---

### 3.1 Overview

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly. A good signal connection can avoid unnecessary and costly damage to your PC and other hardware devices. This chapter provides useful information about how to connect input and output signals to the MIC-3716 via the I/O connector.

### 3.2 I/O Connector

The I/O connector on the MIC-3716 is a 68-pin connector that enables you to connect to accessories with the PCL-10168 shielded cable.

---

**Note:**

The PCL-10168 shielded cable is especially designed for the MIC-3716 to reduce noise in the analog signal lines. Please refer to *1.4 Accessories*.

---

### Pin Assignment

*Fig. 3-1* shows the pin assignments for the 68-pin I/O connector on the MIC-3716.

---

**Note:**

The three ground references AIGND, AOGND, and DGND should each be used discreetly according to their designated purpose. An individual GND pin is available for AI, AO and DIO to provide optimum signal quality.

Since all the signals on the DAS card need to refer to the same GND, the manufacturer has tested and chosen the best point to connect AIGND, AOGND and DGND together. This is based on the "single-point" ground principle.

---

AI0	68	34	AI1
AI2	67	33	AI3
AI4	66	32	AI5
AI6	65	31	AI7
AI8	64	30	AI9
AI10	63	29	AI11
AI12	62	28	AI13
AI14	61	27	AI15
AIGND	60	26	AIGND
*DA0_REF	59	25	DA1_REF*
*DA0_OUT	58	24	DA1_OUT*
*AOGND	57	23	AOGND*
DI0	56	22	DI1
DI2	55	21	DI3
DI4	54	20	DI5
DI6	53	19	DI7
DI8	52	18	DI9
DI10	51	17	DI11
DI12	50	16	DI13
DI14	49	15	DI15
DGND	48	14	DGND
DO0	47	13	DO1
DO2	46	12	DO3
DO4	45	11	DO5
DO6	44	10	DO7
DO8	43	9	DO9
DO10	42	8	DO11
DO12	41	7	DO13
DO14	40	6	DO15
DGND	39	5	DGND
CNT0_CLK	38	4	PACER_OUT
CNT0_OUT	37	3	TRG_GATE
CNT0_GATE	36	2	EXT_TRG
+12V	35	1	+5V

**Fig. 3-1: I/O connector pin assignments for the MIC-3716**

## I/O Connector Signal Description

**Table 3-1: I/O Connector Signal Description**

Signal Name	Reference	Direction	Description
AI<0... 15>	AIGND	Input	<b>Analog Input Channels 0 through 15.</b> Each channel pair, AI<i, i+1> (i = 0, 2, 4...14), can be configured as either two single-ended inputs or one differential input.
AIGND	-	-	<b>Analog Input Ground.</b> The three ground references (AIGND, AOGND, and DGND) are connected together.
A00_REF A01_REF	AOGND	Input	<b>Analog Output Channel 0/1 External Reference.</b>
A00_OUT A01_OUT	AOGND	Output	<b>Analog Output Channels 0/1.</b>
AOGND	-	-	<b>Analog Output Ground.</b> The analog output voltages are referenced to these nodes. The three ground references (AIGND, AOGND, and DGND) are connected together.
DI<0..15>	DGND	Input	<b>Digital Input channels.</b>
DO<0..15>	DGND	Output	<b>Digital Output channels.</b>
DGND	-	-	<b>Digital Ground.</b> This pin supplies the reference for the digital channels at the I/O connector as well as the +5VDC supply. The three ground references (AIGND, AOGND, and DGND) are connected together.
CNT0_CLK	DGND	Input	<b>Counter 0 Clock Input.</b> The clock input of counter 0 can be either external (up to 10 MHz) or internal (1 MHz), as set by the software.
CNT0_OUT	DGND	Output	<b>Counter 0 Output.</b>
CNT0_GATE	DGND	Input	<b>Counter 0 Gate Control.</b>
PACER_OUT	DGND	Output	<b>Pacer Clock Output.</b> This pin pulses once for each pacer clock when turned on. If A/D conversion is in the pacer trigger mode, you can use this signal as a synchronous signal for other applications. A low- to- high edge triggers the A/D conversion to start.
TRG_GATE	DGND	Input	<b>A/D External Trigger Gate.</b> When TRG_GATE is connected to +5 V, it will enable the external trigger signal to input. When TRG_GATE is connected to DGND, it will disable the external trigger signal to input.
EXT_TRG	DGND	Input	<b>A/D External Trigger.</b> This pin is an external trigger signal input for the A/D conversion. A low-to-high edge triggers A/D conversion to start.
+12V	DGND	Output	<b>+12 VDC Source.</b>
+5V	DGND	Output	<b>+5 VDC Source.</b>

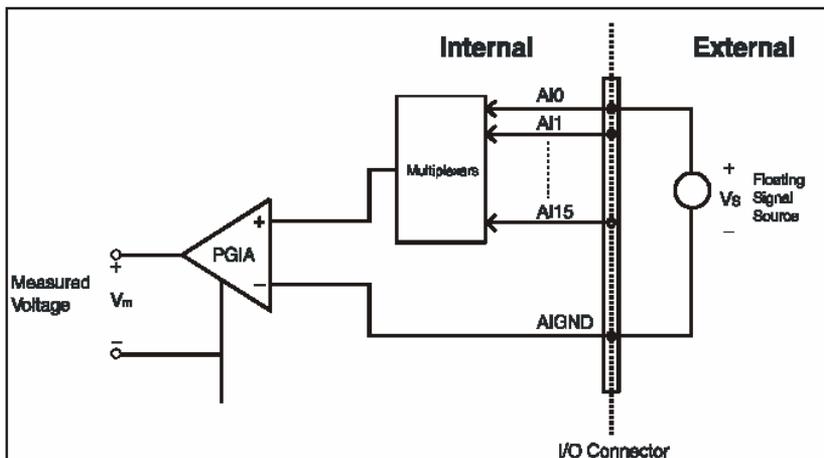
### 3.3 Analog Input Connections

The MIC-3716 supports both 16-channel single-ended or 8 differential analog inputs. Each individual input channel is software-selectable.

#### Single-ended Channel Connections

The single-ended input configuration has only one signal wire for each channel, and the measured voltage ( $V_m$ ) is the voltage of the wire as referenced against the common ground.

A signal source without a local ground is also called a “floating source”. It is fairly simple to connect a single-ended channel to a floating signal source. In this mode, the MIC-3716 provides a reference ground for external floating signal sources. *Fig. 3-2* shows a single-ended channel connection between a floating signal source and an input channel on the MIC-3716.



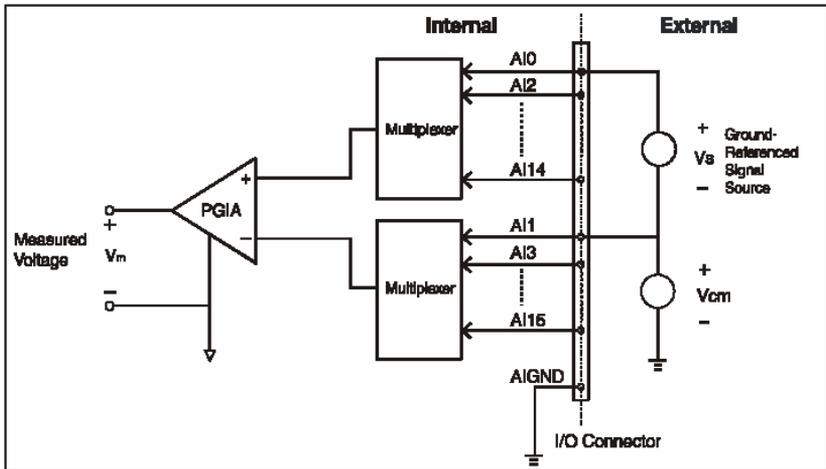
*Fig. 3-2: Single-ended input channel connection*

## Differential Channel Connections

The differential input channels operate with two signal wires for each channel, and the voltage difference between both signal wires is measured. On the MIC-3716, when all channels are configured to differential input, up to 8 analog channels are available.

If one side of the signal source is connected to a local ground, the signal source is ground-referenced. Therefore, the ground of the signal source and the ground of the card will not be exactly of the same voltage. The difference between the ground voltages forms a common-mode voltage ( $V_{cm}$ ).

To avoid the ground loop noise effect caused by common-mode voltages, you can connect the signal ground to the Low input. *Fig. 3-3* shows a differential channel connection between a ground-reference signal source and an input channel on the MIC-3716. With this connection, the PGIA rejects a common-mode voltage  $V_{cm}$  between the signal source and the MIC-3716 ground, shown as  $V_{cm}$  in *Fig. 3-3*.

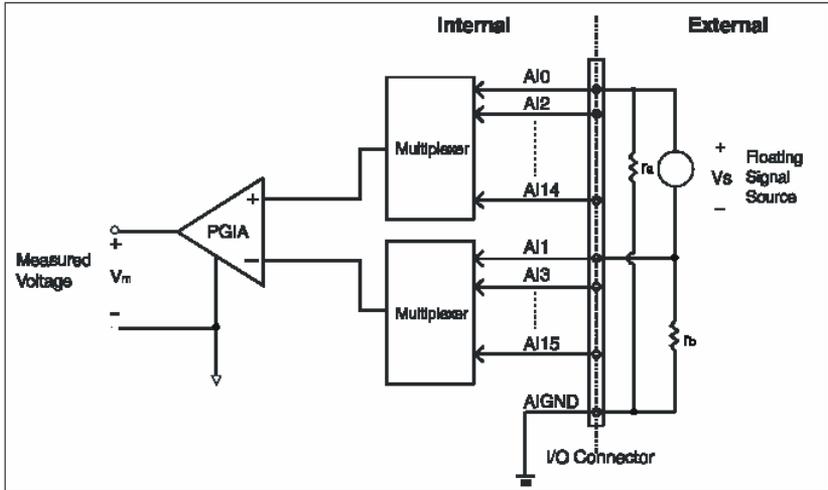


***Fig. 3-3: Differential input channel connection - ground reference signal source***

If a floating signal source is connected to the differential input channel, the signal source might exceed the common-mode signal range of the PGIA, and the PGIA will be saturated with erroneous voltage-readings. You must therefore reference the signal source against the AIGND.

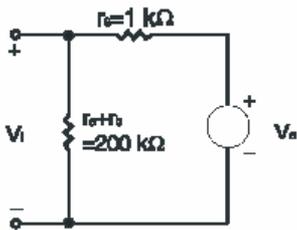
*Fig. 3-4* shows a differential channel connection between a floating

signal source and an input channel on the MIC-3716. In this figure, each side of the floating signal source is connected through a resistor to the AIGND. This connection can reject the common-mode voltage between the signal source and the MIC-3716 ground.



***Fig. 3-4: Differential input channel connection - floating signal source***

This connection has the disadvantage of loading the source down with the series combination (sum) of the two resistors. For  $r_a$  and  $r_b$ , for example, if the input impedance  $r_s$  is 1 k $\Omega$ , and each of the two resistors are 100 k $\Omega$ , then the resistors load down the signal source with 200 k $\Omega$  (100 k $\Omega$  + 100 k $\Omega$ ), resulting in a -0.5% gain error. The following gives a simplified representation of the circuit and calculating process.



$V_s$ : ideal signal source

$V_L$ : measured signal source

$\Gamma_s$ : output impedance of signal source

$\Gamma_s, \Gamma_L$ : series-wound resistors

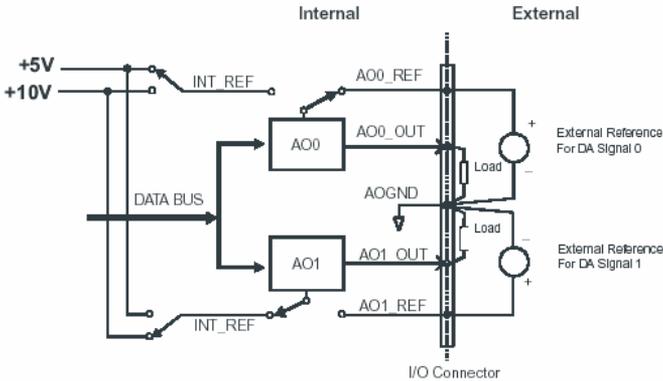
$$V_L = \frac{r_s + \Gamma_L}{r_s + \Gamma_s + \Gamma_L} V_s = \frac{200}{1 + 200} V_s = \frac{200}{201} V_s$$

$$\text{Gain error} = \frac{V_L - V_s}{V_s} = -\frac{1}{201} = -0.5\%$$

### 3.4 Analog Output Connections

The MIC-3716 provides two D/A output channels, **AO0\_OUT** and **AO1\_OUT**. You may use the MIC-3716 internally-provided precision -5V (-10V) reference to generate a 0 to +5 V (+10 V) D/A output range. You may also create a D/A output range through external references, **AO0\_REF** and **AO1\_REF**. The external reference input range is +/-10 V. For example, connecting with an external reference of -7 V will generate a 0 ~ +7 V D/A output.

*Fig. 3-5* shows how to make an analog output and external reference input connections on the MIC-3716.



***Fig. 3-5: Analog output connections***

## 3.5 Trigger Source Connections

### Internal Pacer Trigger Connection

The MIC-3716 includes one 82C54 compatible programmable Timer/Counter chip which provides three 16-bit counters connected to a 10 MHz clock, each designated specifically as Counter 0, Counter 1 and Counter 2. Counter 0 is a counter which counts events from an input channel or output pulse. Counter 1 and Counter 2 are cascaded to create a 32-bit timer for pacer triggering. A low-to-high edge from the Counter 2 output (**PACER\_OUT**) will trigger an A/D conversion on the MIC-3716. At the same time, you can also use this signal as a synchronous signal for other applications.

### External Trigger Source Connection

In addition to pacer triggering, the MIC-3716 also allows external triggering for A/D conversions. When a +5 V source is connected to **TRG\_GATE**, the external trigger function is enabled. A low-to-high edge coming from **EXT\_TRG** will trigger an A/D conversion on the MIC-3716. When **DGND** is connected to **TRG\_GATE**, the external trigger function is thereby disabled.

## 3.6 Field Wiring Considerations

When you use the MIC-3716 to acquire data from outside, noises in the environment might significantly affect the accuracy of your measurements if cautions are not taken. The following measures will help reduce possible interference running signal wires between signal

sources and the MIC-3716.

- ◆ The signal cables must be kept away from strong electromagnetic sources such as power lines, large electric motors, circuit breakers or welding machines, since they may cause strong electromagnetic interference. Keep the analog signal cables away from any video monitor, since it can significantly affect a data acquisition system.
- ◆ If the cable travels through an area with significant electromagnetic interference, you should adopt individually shielded, twisted-pair wires as the analog input cable. This type of cable has its signal wires twisted together and shielded with a metal mesh. The metal mesh should only be connected to one point at the signal source ground.
- ◆ Avoid running the signal cables through any conduit that might have power lines in it.
- ◆ If you have to place your signal cable parallel to a power line that has a high voltage or high current running through it, try to keep a safe distance between them. Or, you should place the signal cable at a right angle to the power line to minimize undesirable effects.
- ◆ The signals transmitted on the cable will be directly affected by the quality of the cable. In order to ensure better signal quality, we recommend that you use the PCL-10168 shielded cable available from Advantech.

*(This page is left blank for hard printing.)*

CHAPTER

4

**Software  
Programming  
Overview**

## 4. Software Programming Overview

---

This chapter gives you an overview of the software programming choices available and a quick reference to source code examples that can help you during programming. After following the instructions given in *Chapter 2*, it is hoped that you feel comfortable enough to proceed further.

Programming choices for DAS cards:

You may use Advantech application software such as the Advantech DLL driver to program your DAS cards. Advanced users may also wish to choose register-level programming, although this is not recommended due to its laborious and time-consuming nature.

### 4.1 Programming Choices

#### DLL Driver

The Advantech DLL Drivers software is included on the companion CD-ROM. It also comes with all Advantech DAS cards. Advantech's DLL driver features a complete I/O function library to help boost your application performance. The Advantech DLL driver for *Windows 95/98/NT/2000/XP* works seamlessly with development tools such as Visual C++, Visual Basic, Borland C++ Builder and Borland Delphi.

#### Register-level Programming

Register-level programming is reserved for experienced programmers who find it necessary to write codes directly at the level of device registers. Since register-level programming requires much effort and time, we recommend that you use the Advantech DLL drivers instead. However, if register-level programming is indispensable, you should refer to the relevant information in *Appendix C, Register Structure and Format*, and/or to the example codes included on the companion CD-ROM.

## 4.2 DLL Driver Programming Roadmap

This section will provide a roadmap to demonstrate how to build an application from scratch using the Advantech DLL driver with your favorite development tools such as Visual C++, Visual Basic, Delphi and C++ Builder. The step-by-step instructions on how to build your own applications using each development tool will be given in the *DLL Drivers Manual*. Moreover, a rich set of example source codes are also given for your reference.

### Programming Tools

Programmers can develop application programs with their favorite development tools:

- ◆ **Visual C++**
- ◆ **Visual Basic**
- ◆ **Delphi**
- ◆ **C++ Builder**

For instructions on how to begin programming in each development

tool, Advantech offers a *Tutorial* Chapter in the *DLL Drivers Manual* for your reference. Please refer to the corresponding sections in this chapter on the *DLL Drivers Manual* to begin your programming efforts. You can also take a look at the example source codes provided for each programming tool, since they can prove very helpful.

The *DLL Drivers Manual* can be found on the companion CD-ROM. Or if you have already installed the DLL Drivers on your system, The *DLL Drivers Manual* can be readily accessed through the **Start** button: **Start/Programs/Advantech Automation/Device Manager/Device Driver's Manual**

The example source codes should be found under the corresponding installation folder such as the default installation path:

***\Program Files\ADVANTECH\ADSAPI\Examples***

For information about using other function groups or other development tools, please refer to the *Creating Windows95/98/NT/2000/XP Application with DLL Driver* chapter and the *Function Overview* chapter on the *DLL Drivers Manual*.

## **Programming with the DLL Driver Function Library**

The Advantech DLL driver offers a rich function library to be utilized in various application programs. This function library consists of

numerous APIs that support many development tools, such as Visual C++, Visual Basic, Delphi and C++ Builder.

According to their specific functions or services, the APIs can be categorized into several function groups:

- ◆ **Analog Input Function Group**
- ◆ **Analog Output Function Group**
- ◆ **Digital Input/Output Function Group**
- ◆ **Counter Function Group**
- ◆ **Temperature Measurement Function Group**
- ◆ **Alarm Function Group**
- ◆ **Port Function Group**
- ◆ **Communication Function Group**
- ◆ **Event Function Group**

For the usage and parameters of each function, please refer to the *Function Overview* chapter in the *DLL Drivers Manual*.

## **Troubleshooting DLL Driver Error**

Driver functions will return a status code when they are called to perform a certain task for the application. When a function returns a code that is not zero, it means the function has failed to perform its designated function. To troubleshoot the DLL driver error, you can pass the error code to the **DRV\_GetErrorMessage** function to return the error message. Or you can refer to the *DLL Driver Error Codes Appendix* in the *DLL Drivers Manual* for a detailed listing of the Error Code, Error ID and the Error Message.

*(This page is left blank for hard printing.)*

CHAPTER

5

**Calibration**

# 5. Calibration

---

## 5.1 Overview

This chapter provides a brief overview on MIC-3716 calibration.

Regular calibration checks are important to maintain accuracy in data acquisition and control applications. We provide calibration programs and utilities on the companion CD-ROM to assist you in A/D and D/A calibration.

---

**Note:**

If you installed the program to another directory, you can find these programs in the corresponding subfolders in your destination directory.

---

The MIC-3716 has been calibrated at the factory for initial use. However, a calibration of the analog input and the analog output functions every six months is recommended.

These calibration programs make calibration an easy job. With a variety of prompts and graphic displays, these programs will lead you through the calibration and setup procedures, showing you all the correct settings and adjustments.

To perform a satisfactory calibration, you will need a 4½-digit digital multi-meter, and a voltage calibrator or a stable, noise-free DC voltage source.

---

**Note:**

Before you calibrate the A/D or D/A function, power must have been turned on for at least **15 minutes** to make sure the DAS card is stable.

---

A calibration utility, *AutoCali.EXE*, is included on the companion CD-ROM :

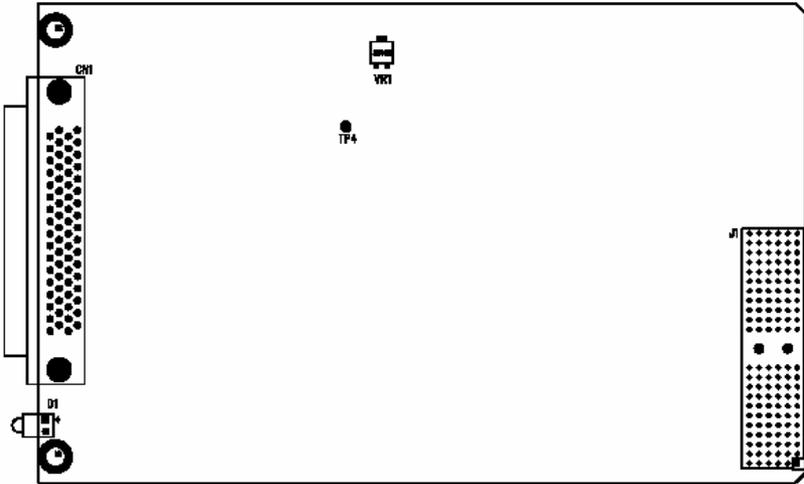
***AutoCali.EXE***      MIC-3716 calibration utility

This calibration utility is designed for the Microsoft Windows environment. You can access this program from the default location:

***\Program Files\ADVANTECH\ADSAPI\Utilities\MIC3716***

## **5.2 VR Assignment**

There is one variable resistor (VR1) on the MIC-3716 to adjust the accurate reference voltage on the MIC-3716. We have provided a test point (See TP4 in *Fig. 5-1*) for you to check the reference voltage on board. Before you start to calibrate A/D and D/A channels, please adjust VR1 until the reference voltage on TP4 has reached +5.0000 V. *Fig. 5-1* shows the locations of VR1 and TP4.



*Fig. 5-1: MIC-3716 VR assignment*

### 5.3 Auto-Calibration

The calibration utility, *AutoCali.EXE*, provides four functions - auto A/D calibration, auto D/A calibration, manual A/D calibration and manual D/A calibration. The program helps the user to easily finish the calibration procedures automatically; however, the user can calibrate the MIC-3716 manually. Appendix E illustrates the standard calibration procedure for your reference. If you want to calibrate the hardware in your own way, these two sections will guide you. The following steps will guide you through the MIC-3716 software calibration.

**Step 1:** Access the calibration utility program *AutoCali.exe* from the default location:

*C: \Program Files\ADVANTECH\ADSAPI\Utilities\*

## MIC3716

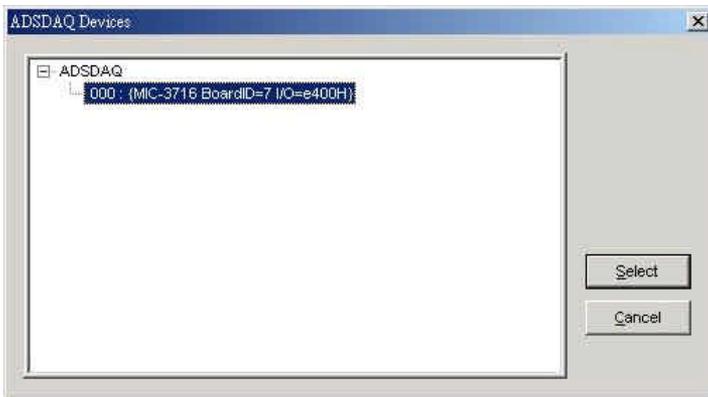
---

**Note:**

If you installed the program to another directory, you can find this program in the corresponding subfolders in your destination directory.

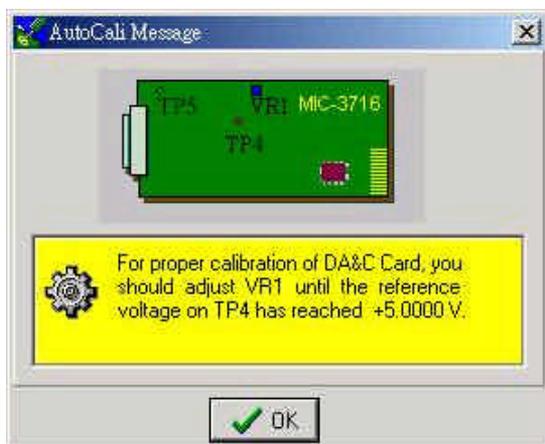
---

**Step 2:** Select MIC-3716 in the ADSDAQ dialog box.



*Fig. 5-2: Selecting the device you want to calibrate*

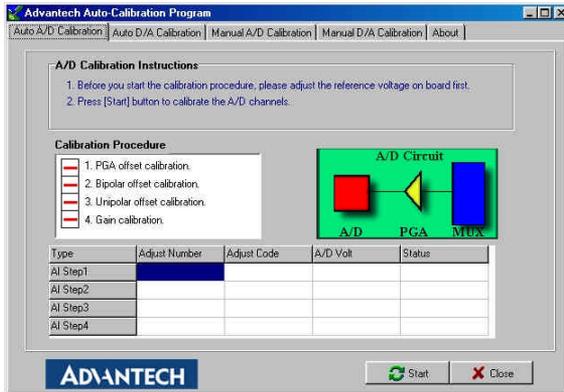
**Step 3:** After you start to calibrate the MIC-3716, please don't forget to adjust VR1.



*Fig. 5-3: Warning message before start calibration*

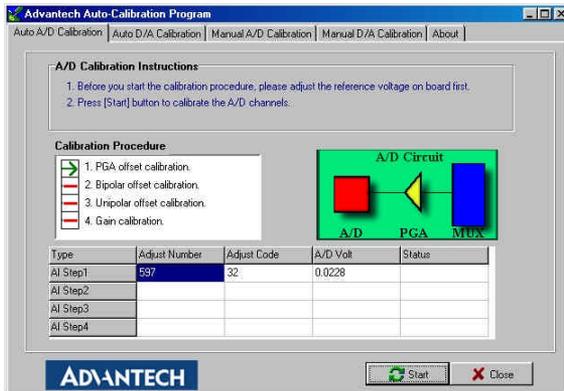
## A/D channel Auto-Calibration

**Step 4:** Click the *Auto A/D Calibration* tab to show the A/D channel auto-calibration panel (Fig. 5-4). Press the start button to calibrate A/D channels automatically.



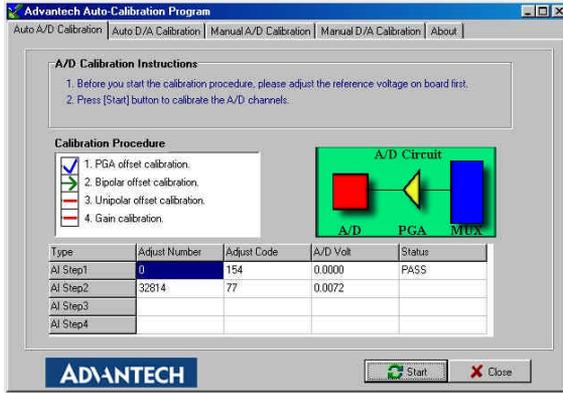
**Fig. 5-4: Auto A/D Calibration Dialog Box**

**Step 5:** The first A/D calibration procedure is enabled (Fig. 5-5).



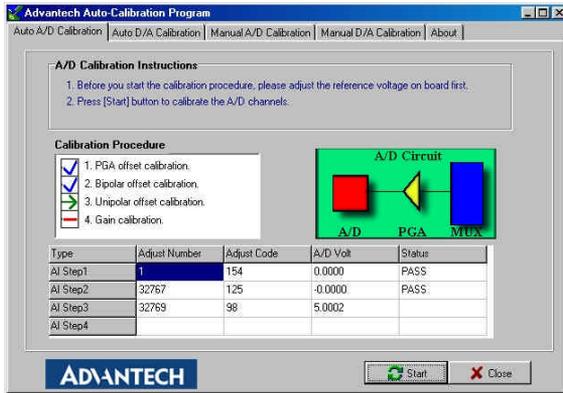
**Fig. 5-5: A/D Calibration Procedure 1**

**Step 6:** The second A/D calibration procedure is enabled (Fig. 5-6)



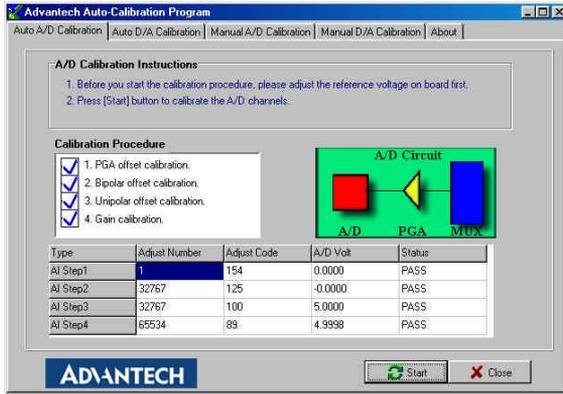
**Fig. 5-6: A/D Calibration Procedure 2**

**Step 7:** The third A/D calibration procedure is enabled (Fig. 5-7)



**Fig. 5-7: A/D Calibration Procedure 3**

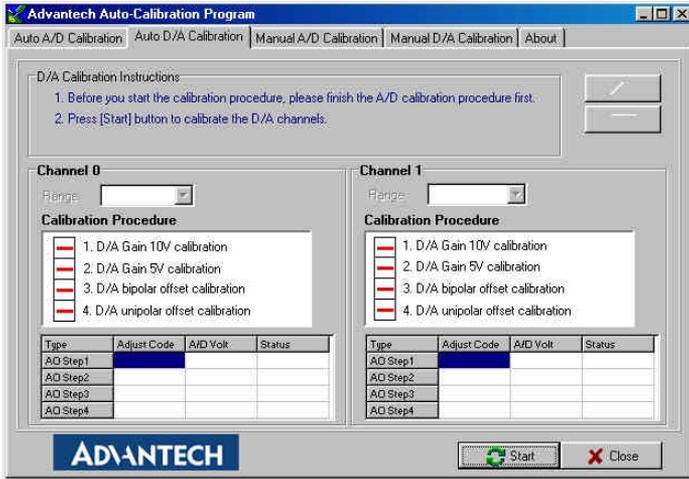
**Step 8:** Auto-calibration is finished. (Fig. 5-8)



**Fig. 5-8: A/D Calibration is finished**

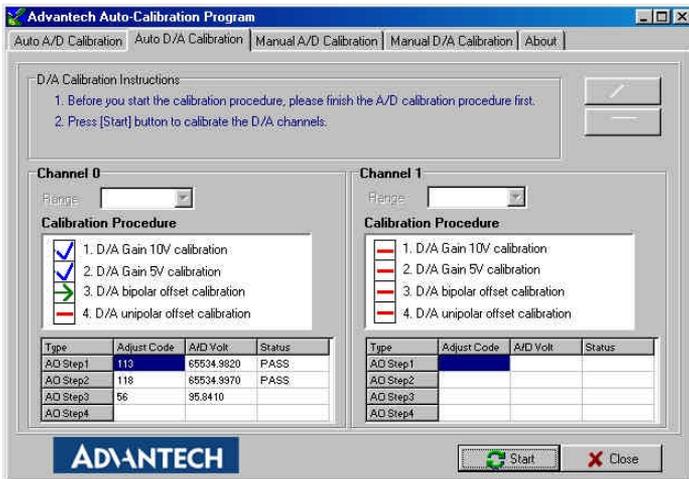
## D/A channel Auto-Calibration

**Step 9:** Click the *Auto D/A Calibration* tab to show the D/A channel auto calibration panel. Please finish the A/D calibration procedure first before you start the D/A calibration procedure. There are two D/A channels in MIC-3716; select the output range for each channel and then press the start button to calibrate the D/A channels (Fig. 5-9).



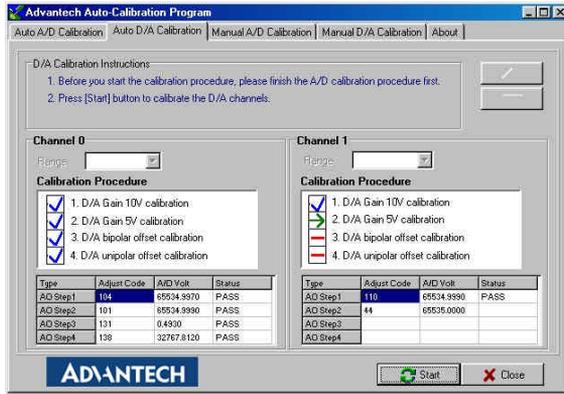
**Fig. 5-9: Range Selection in D/A Calibration**

**Step 10:** D/A channel 0 calibration is enabled (Fig. 5-10)



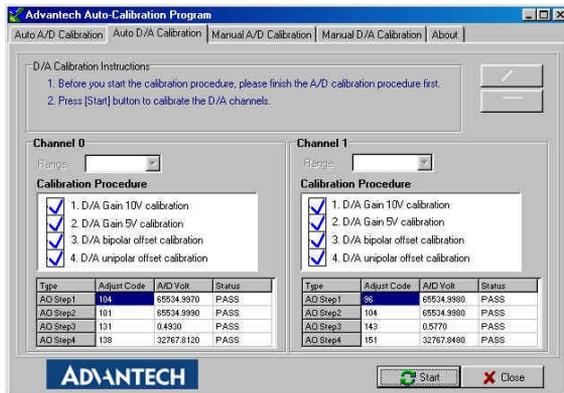
**Fig. 5-10: Calibrating D/A Channel 0**

**Step 11: D/A channel 1 calibration is enabled (Fig. 5-11)**



**Fig. 5-11: Calibrating D/A Channel 1**

**Step 12: Auto-calibration is finished (Fig. 5-12)**



**Fig. 5-12: D/A Calibration is finished**

## 5.4 Manual Calibration

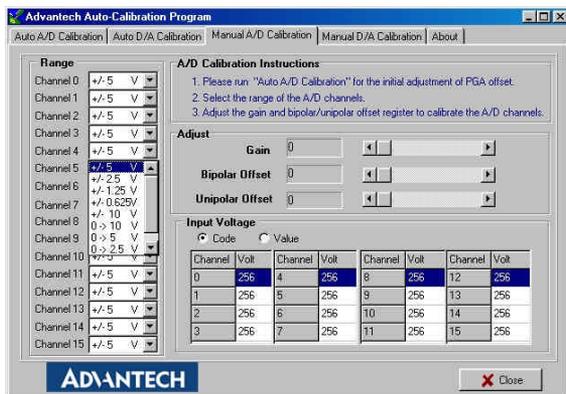
### A/D channel Manual-Calibration

**Step 1:** Click the *Manual A/D Calibration* tab to show the A/D channel manual calibration panel. Before calibrating, acquire the reference voltage from a precision standard voltage reference. Go to the **Range** form, select a channel and the target voltage range according to the input voltage value from a precision standard voltage reference (*Fig. 5-13*).

---

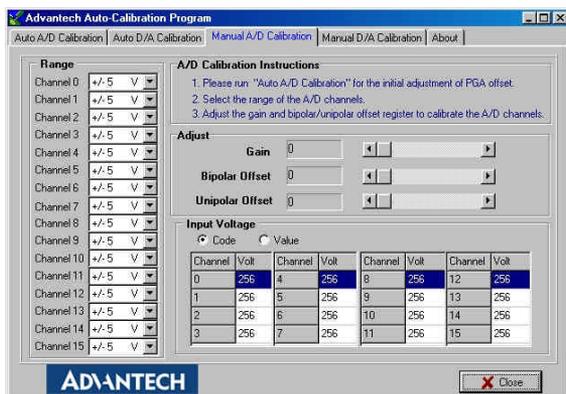
**Note:**

- The input voltage value you selected from a precision standard voltage reference needs to correspond with the one that the MIC-3716 can read.
  - The input voltage will be analog so the computer will convert the voltage to a digital voltage; therefore, the input voltage value you selected from a precise standard voltage reference needs to correspond with the one that the MIC-3716 can read. For example, if the input range is 0 ~ 5V, the input voltage should be 2.9992V not 3V.
-



**Fig. 5-13: Selecting Input Range in the Manual A/D Calibration panel**

**Step 2:** Depending on the difference between reference voltage and the received data in MIC-3716, adjust the gain, bipolar offset and unipolar offset registers (Fig. 5-14)



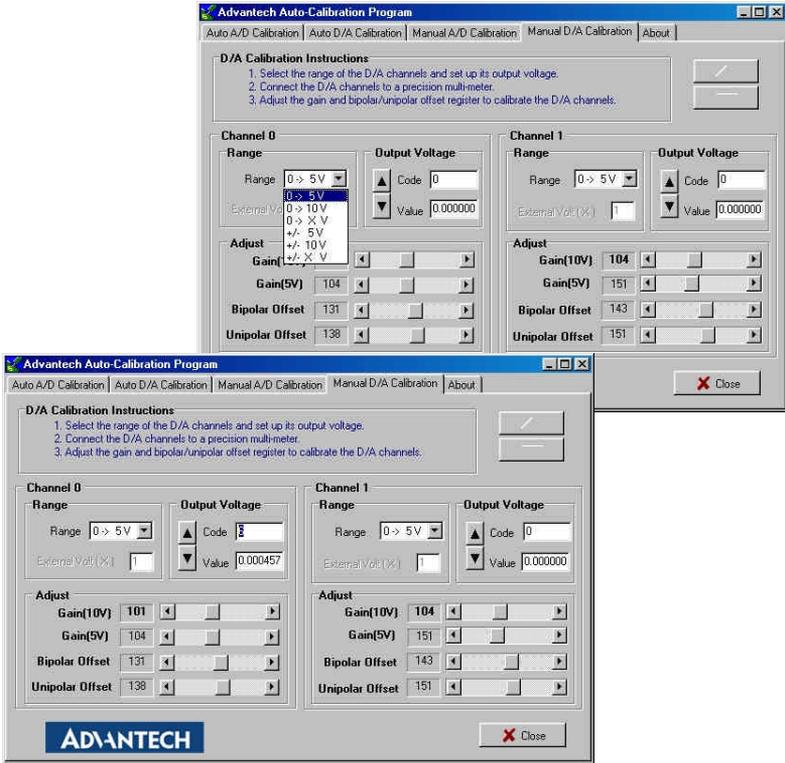
**Fig. 5-14: Adjusting registers**

**Step 3:** Adjust the registers until they fall between the input voltage from the standard voltage reference and the receiving voltage reflected in the Manual A/D Calibration tab.

## **D/A channel Manual-Calibration**

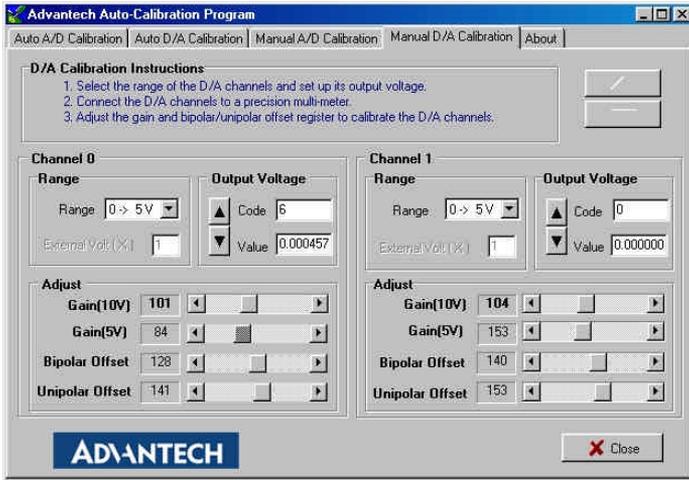
**Step 1:** Click the *Manual D/A Calibration* tab to show the D/A channel manual calibration panel. The two D/A channels are individually calibrated. Before calibrating, create your desired output voltage from the D/A channels and measure it through an external precision multi-meter.

**Step 2:** For example, choose channel 0; select the Range and select the wished output voltage code or value from the radio buttons (Fig. 5-15 and Fig. 5-16).



**Fig. 5-15 & Fig. 5-16: Selecting D/A Range and Choosing Output Voltage**

**Step 3:** According to the difference between the output voltage from D/A channel and the value in the multi-meter, adjust the gain, bipolar offset and unipolar offset registers (*Fig. 5-17*)



*Fig. 5-17: Adjusting registers*

**Step 4:** Adjust registers until they fall between the output voltage from the D/A channel and the value in the multi-meter.

## **Appendixes**

# Appendix A. Specifications

## Analog Input

Channels	16 Single-Ended or 8 differential or combination					
Resolution	16-bit					
FIFO Size	1K samples					
Max. Sampling Rate	250 KS/s max.					
Conversion Time	2.5 $\mu$ s					
Input range and Gain List	Gain	0.5	1	2	4	8
	Unipolar	N/A	0-10	0-5	0-2.5	0-1.25
	Bipolar	$\pm 10V$	$\pm 5V$	$\pm 2.5V$	$\pm 1.25V$	$\pm 0.625V$
Small Signal Bandwidth for PGA	Gain	0.5	1	2	4	8
	Bandwidth	4.0 M Hz	4.0 M Hz	2.0 M Hz	1.5 M Hz	0.65 M Hz
Common mode Voltage	$\pm 11$ V max. (operational)					
Max. Input Voltage	$\pm 20V$					
Input Protect	30 Vp-p					
Input Impedance	100 M $\Omega$ / 10 pF(Off) ; 100 M $\Omega$ / 10 OpF(On)					
Trigger Mode	Software, On-board Programmable Pacer or external					
Accuracy	DC	INLE: $\pm 1$ LSB				
		DNLE: $\pm 1$ LSB				
		Zero (Offset) error: Adjustable to $\pm 1$ LSB				
		Gain	0.5	1	2	4
	Gain error (%FSR)	0.03	0.02	0.02	0.03	0.04
	AC	SNR : 82 dB				
		ENOB: 13.5 bits				
Clocking and Trigger Inputs	Trigger Mode	Software, on-board programmable pacer or external				
	A/D pacer clock	250 k Hz (max.) ; 58 $\mu$ s Hz (min.)				
	External A/D trigger Clock	Min. pulse width: 2 $\mu$ s (high); 2 $\mu$ s (low) Max. frequency: 250kHz				

## Analog Output

Channels	2	
Resolution	16-bit	
Operation mode	Single output	
Throughput	200 KS/s max. per channel (FSR)	
Output Range (Internal & External Reference)	Using Internal Reference	0 - +5V, 0 - +10V, -5V - +5V, -10V - +10V
	Using External Reference	0 - +x V @ + x V ( - 10 x 10 ) - x - +x V @ + x V ( - 10 x 10 )
Accuracy	DC	DNLE: ± 1 LSB (monotonic)
		INLE: ± 1 LSB
		Zero (Offset) error: Adjustable to ± 1 LSB
		Gain (Full-scale) error: Adjustable to ± 1 LSB
Dynamic Performance	Setting Time	5 µs (to 4 LSB of FSR)
	Slew Rate	20 V / µs
Drift	10 ppm / °C	
Driving Capability	±20mA	
Output Impedance	0.10 max.	

## Digital Input/Output

Input Channels	16	
Input Voltage	Low	0.4 V max.
	High	2.4V min.
Input Load	Low	0.4 V max. @ 0.2mA
	High	2.7V min. @ 20µ A
Output Channels	16	
Output Voltage	Low	0.4 V max. @ + 8.0 mA (sink)
	High	2.4V min. @ - 0.4 mA (source)

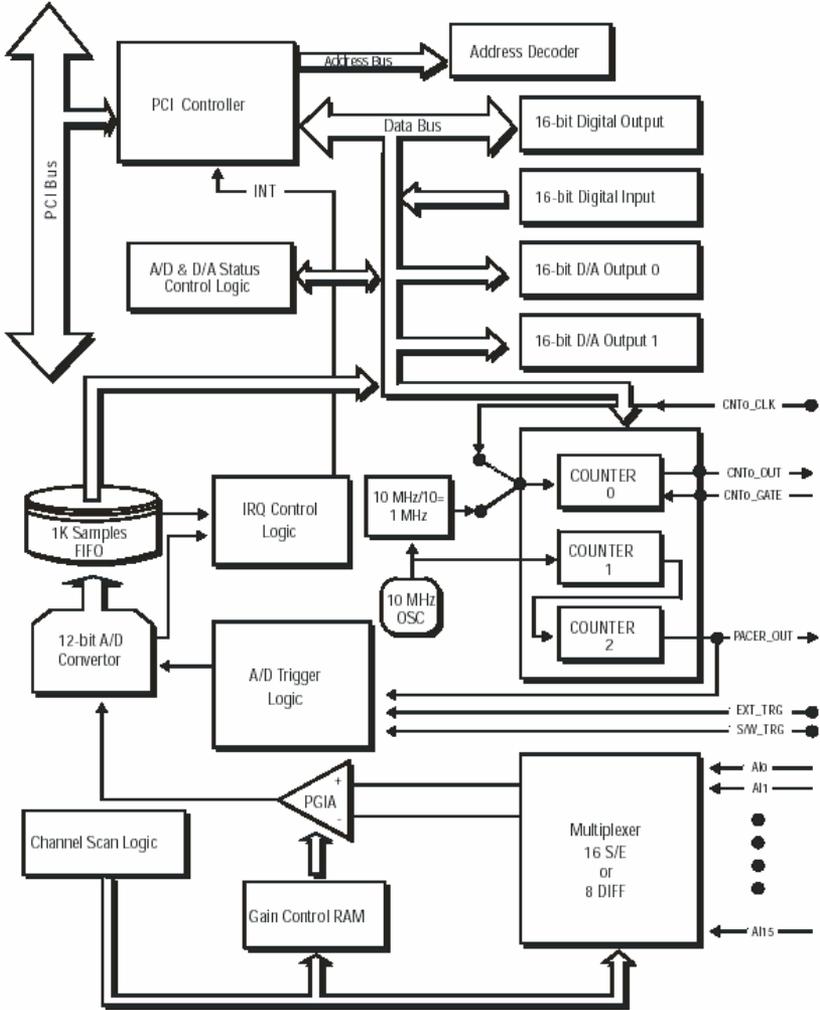
## Counter/Timer

Channels	3 channels, 2 channels are permanently configured as programmable pacers;1 channel is free for user application	
Resolution	16-bit	
Compatibility	TTL level	
Base Clock	Channel 2:Takes input from output of channel 1 Channel 1:1MHz Channel 0: Internal 1MHz or external clock ( 10 MHz ) max Selected by software	
Max. Input Frequency	1 M Hz	
Clock Input	Low	0.8 V max.
	High	2.0 V min.
Gate Input	Low	0.8 V max.
	High	2.0 V min.
Counter Output	Low	0.5 V max. @ +24mA
	High	2.4 V min. @ -15mA

## General

<b>I/O Connector Type</b>	68-pin SCSI-II female	
<b>Dimensions</b>	175 mm x 100 mm ( 6.9" x 3.9" )	
<b>Power Consumption</b>	<b>Typical</b>	+ 5 V @ 850mA + 12V @ 600mA
	<b>Max.</b>	+ 5 V @ 1 A + 12V @ 700mA
<b>Temperature</b>	<b>Operation</b>	0 ~ +60°C ( 32~ 158 ) (refer to IEC 65 – 2 - 1 ,2)
	<b>Storage</b>	-20 ~ +85°C ( -4 ~158 )
<b>Relative Humidity</b>	<b>Operation</b>	5 ~ 85% RH non-condensing ( refer to IEC 68 -1,-2,-3)
	<b>Storage</b>	5 ~ 95% RH non-condensing ( refer to IEC 68 -1,-2,-3)
<b>Certification</b>	CE certified	

# Appendix B. Block Diagrams



*(This page is left blank for hard printing.)*

# Appendix C. Register Structure and Format

---

## C.1 Overview

The MIC-3716 is delivered with an easy-to-use 32-bit DLL driver for user programming under the *Windows 95/98/NT/2000/XP* operating system. We advise users to program the MIC-3716 using the 32-bit DLL driver provided by Advantech to avoid the complexity of low-level programming by registers.

The most important consideration in programming the MIC-3716 at the register level is to understand the function of the card's registers. The information in the following sections is provided only for users who would like to do their own low-level programming.

## C.2 I/O Port Address Map

The MIC-3716 requires 32 consecutive addresses in the PC's I/O space. The address of each register is specified as an offset from the card's base address. For example, BASE+0 is the card's base address and BASE+7 is the base address plus seven bytes.

*Table C-1* shows the function of each register of the MIC-3716 or driver and its address relative to the card's base address.

**Table C-1: MIC-3716 register format (Part 1)**

Base Address +decimal	Read							
	7	6	5	4	3	2	1	0
	<b>A/D Data</b>							
1	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8
0	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
	<b>N/A</b>							
3								
2								
	<b>N/A</b>							
5								
4								
	<b>A/D Status Register</b>							
7	CAL				IRQ	F/F	F/H	F/E
6	AD16/12	CNT0	ONE/FH	IRQEN	GATE	EXT	PACER	SW
	<b>N/A</b>							
9								
8								
	<b>D/A channel 0 data</b>							
11								
10								
	<b>D/A channel 1 data</b>							
13								
12								
	<b>N/A</b>							
15								
14								

**Table C-1: MIC-3716 register format (Part 2)**

Base Address +decimal	Read							
	7	6	5	4	3	2	1	0
	<b>Digital Input</b>							
17	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
16	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
	<b>N/A</b>							
19								
18								
	<b>Board ID</b>							
21								
20					BD3	BD2	BD1	BD0
	<b>N/A</b>							
23								
22								
	<b>Counter 0</b>							
25								
24	D7	D6	D5	D4	D3	D2	D1	D0
	<b>Counter 1</b>							
27								
26	D7	D6	D5	D4	D3	D2	D1	D0
	<b>Counter 2</b>							
29								
28	D7	D6	D5	D4	D3	D2	D1	D0
	<b>N/A</b>							
31								
30								

**Table C-1: MIC-3716 register format (Part 3)**

Base Address +decimal	Write							
	7	6	5	4	3	2	1	0
	<b>Software A/D Trigger</b>							
1								
0								
	<b>A/D Channel Range Setting</b>							
3								
2								
	<b>Multiplexer Control</b>							
5					Stop channel			
4					Start channel			
	<b>A/D Control Register</b>							
7	CAL							
6	AD16/12	CNT0	ONE/FH	IROEN	GATE	EXT0	PACER	SW
	<b>Clear Interrupt and FIFO</b>							
9	Clear FIFO							
8	Clear interrupt							
	<b>D/A Output Channel 0</b>							
11	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
10	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
	<b>D/A Output Channel 1</b>							
13	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
12	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
	<b>D/A Control Register</b>							
15					DA1_LDEN	DA1_I/E	DA0_B/U	DA1_5/10
14					DA0_LDEN	DA0_I/E	DA0_B/U	DA0_5/10

**Table C-1: MIC-3716 register format (Part 4)**

Base Address +decimal	Write							
	7	6	5	4	3	2	1	0
	<b>Digital Output</b>							
17	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8
16	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
	<b>Calibration Command and Data</b>							
19					CM3	CM2	CM1	CM0
18	D7	D6	D5	D4	D3	D2	D1	D0
	<b>N/A</b>							
21								
20								
	<b>N/A</b>							
23								
22								
	<b>Counter 0</b>							
25								
24								
	<b>Counter 1</b>							
27								
26								
	<b>Counter 2</b>							
29								
28								
	<b>Counter Control</b>							
31								
30								

### C.3 Channel Number and A/D Data — BASE+0 and BASE+1

BASE+0 and BASE+1 hold the result of A/D conversion data. The 16 bits of data from the A/D conversion are stored in BASE+1 bit 7 to bit 0 and BASE+0 bit 7 to bit 0.

**Table C-2: MIC-3716 Register for A/D data**

Read	A/D Data							
Bit #	7	6	5	4	3	2	1	0
BASE + 1	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8
BASE + 0	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

- AD15 ~ AD0**      Result of A/D Conversion
- AD0      the least significant bit (LSB) of A/D data
- AD15      the most significant bit (MSB)

## C.4 Software A/D Trigger — BASE+0

You can trigger an A/D conversion by software, the card's on-board pacer or an external pulse.

BASE+6, Bit 2 to bit 0, select the trigger source.

(see *Section C.7, Control Register -- BASE+6* )

If you select software triggering, a write to the register BASE+0 with any value will trigger an A/D conversion.

### C.5 A/D Channel Range Setting — BASE+2

Each A/D channel has its own input range, controlled by a gain code stored in the on-board RAM.

To change the range code for a channel:

- ◆ Write the same channel in BASE+4 (the start channel) and BASE+5 (the stop channel) (refer to *Section C.6*).
- ◆ Write the gain code to BASE+2 bit 0 to bit 2.

**Table C-3: Register for A/D channel range setting**

Write	A/D Channel Range Setting							
Bit #	7	6	5	4	3	2	1	0
BASE + 2			S/D	B/U		G2	G1	G0

**S/D** Single-ended or Differential

0 single-ended

1 differential.

**B/U** Bipolar or Unipolar

0 bipolar

1 unipolar.

**G2 to G0** Gain Code (listed as *Table C-5*)

**Table C-4: Gain codes for MIC-3716**

Gain	Input Range(V)	B/U	Gain Code		
			G2	G1	G0
1	-5 to +5	0	0	0	0
2	-2.5 to +2.5	0	0	0	1
4	-1.25 to +1.25	0	0	1	0
8	-0.625 to +0.625	0	0	1	1
0.5	-10 to +10	0	1	0	0
	N/A	0	1	0	1
	N/A	0	1	1	0
	N/A	0	1	1	1
1	0 to 10	1	0	0	0
2	0 to 5	1	0	0	1
4	0 to 2.5	1	0	1	0
8	0 to 1.25	1	0	1	1
	N/A	1	1	0	0
	N/A	1	1	0	1
	N/A	1	1	1	0
	N/A	1	1	1	1

Example: To set channel 3 as gain=1

1. Write channel 3 to BASE+4 as 00000011.
2. Write channel 3 to BASE+5 as 00000011.
3. Refer to the gain code list, write gain=1 to BASE+2 as 00000000.

## C.6 MUX Control — BASE+4 and BASE+5

*Table C-5: Register for multiplexer control*

Write	Multiplexer Control							
Bit #	7	6	5	4	3	2	1	0
BASE + 5					STO3	STO2	STO1	STO0
BASE + 4					STA3	STA2	STA1	STA0

**STA3 ~ STA0**      Start Scan Channel Number

**STO3 ~ STO0**      Stop Scan Channel Number

- When you set the gain code of analog input channel n, you should set the Multiplexer start & stop channel number to channel n to prevent any unexpected errors. In fact BASE+4 bit 3 to bit 0, STA3 ~ STA0, act as a pointer to channel n’s address in the SRAM when you program the A/D channel setting (*refer to Section C.5*).

**Caution!**

We recommend you to set the same start and stop channel when writing to the register BASE+2. Otherwise, if the A/D trigger source is on, the multiplexer will continuously scan between channels and the range setting may be set to an unexpected channel. Make sure the A/D trigger source is turned off to avoid this kind of error.

The write-only registers of BASE +4 and BASE+5 control how the multiplexers (Multiplexer) scan.

- ◆ BASE+4 bit 3 to bit 0, STA3 ~ STA0, hold the start scan channel number.
- ◆ BASE+5 bit 3 to bit 0, STO3 ~ STO0, hold the stop scan channel number.

Writing to these two registers automatically initializes the scan range of the Multiplexer. Each A/D conversion trigger also sets the Multiplexer to the next channel. With continuous triggering, the Multiplexer will scan from the start channel to the stop channel and then repeat. The following examples show the scan sequences of the Multiplexer.

### **Example 1**

If the start scan input channel is AI3 and the stop scan input channel is AI7, then the scan sequence is AI3, AI4, AI5, AI6, AI7, AI3, AI4, AI5, AI6, AI7, AI3, AI4...

### **Example 2**

If the start scan channel is AI13 and the stop scan channel is AI2, then the scan sequence is AI13, AI14, AI15, AI0, AI1, AI2, AI13, AI14, AI15, AI0, AI1, AI2, AI13, AI14...

The scan logic of the MIC-3716 card is powerful and easily understood. You can set the gain code, B/U and S/D, for each channel. For the Analog Input function, we set two AI channel AI<i, i+1> (i= 0, 2, 4, ...,14) to work as a pair. For example, the AI0 and AI1 is a pair.

When in single-ended mode, we can get data from AI0 and AI1 separately. But if we set them as differential mode, the results polling AI0 and AI1 will be the same. That is if we set the AI0 and AI1 as a differential input channel, we can get the correct result no matter if we poll channel 0 or channel 1.

But if we want to use the multiple channels input function, things will be a little bit different. If we set two AI channels as differential channels, it will be considered as one channel in the data array. Since the resulted data array of the multi-channel scan function is ranked with the order of channel, let us give an example to make it more clear. Now we set channel 0 and 1 as differential and 2, 3 as single ended and then 4,5 as differential mode. And we set the start channel as channel 0 and number of channels as 4. The result will be:

```
##.#### -> channel 0,1  
##.#### -> channel 2  
##.#### -> channel 3  
##.#### -> channel 4,5  
##.#### -> channel 0,1  
##.#### -> channel 2  
##.#### -> channel 3  
##.#### -> channel 4,5  
##.#### -> channel 0,1
```

---

**Warning!**

Only even channels can be set as differential. An odd channel will become unavailable if its preceding channel is set as differential.

---

## C.7 Control Register — BASE+6 and BASE+7

The write-only register BASE+6 and BASE+7 allows users to set an A/D trigger source and an interrupt source.

**Table C-6: Control Register**

Write	A/D Status Register							
Bit #	7	6	5	4	3	2	1	0
BASE + 7	CAL							
BASE + 6	AD16/12	CNT0	ONE/FH	IRQEN	GATE	EXT	PACER	SW

**SW** Software trigger enable bit

- 1 enable
- 0 disable.

**PACER** Pacer trigger enable bit

- 1 enable
- 0 disable.

**EXT** External trigger enable bit

- 1 enable;
- 0 disable.

**Note:**

You cannot enable SW, PACER and EXT concurrently.

**GATE** External trigger gate function enable bit.

- 0 Disable
- 1 Enable

**IRQEN** Interrupt enable bit.

- 0 Disable
  - 1 Enable
- ONE/FH** Interrupt source bit
- 0 Interrupt when an A/D conversion occurs
  - 1 Interrupt when the FIFO is half full.
- CNT0** Counter 0 clock source select bit
- 0 The clock source of Counter 0 comes from the internal clock, 1 MHz
  - 1 The clock source of Counter 0 comes from the external clock, maximum up to 10 MH
- AD16/12** Analog Input resolution.
- 0 16 bit
  - 1 12 bit. And those two registers BASE+0 & BASE+1 (*Table C-7*)

***Table C-7: MIC-3716 Register for channel number and A/D data***

Read	Channel Number and A/D Data							
Bit #	7	6	5	4	3	2	1	0
BASE + 1	CH3	CH2	CH1	CH0	AD11	AD10	AD9	AD8
BASE + 0	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

- AD11 ~ AD0** Result of A/D Conversion
- AD0 the least significant bit (LSB) of A/D data
  - AD11 the most significant bit (MSB)

**CH3 ~ CH0**

A/D Channel Number

CH3 ~ CH0 hold the number of the A/D channel from which the data is received

CH3 MSB

CH0 LSB

**CAL**

Analog I/O calibration bit

0 Normal mode

All analog input and outputs channels are connected to the 68 pin SCSI-II connector respectively.

1 A/D and D/A calibration mode

The wiring becomes that AI0 is connected to 0 V (AGND), AI2 is connected to +5 V, AI4 is connected to AO0, and AI6 is connected to AO1 automatically.

## C.8 Status Register — BASE+6 and BASE+7

The registers of BASE+6 and BASE+7 provide information for A/D configuration and operation.

*Table C-8: Status Register*

Write	A/D Control Register							
Bit #	7	6	5	4	3	2	1	0
BASE + 7	CAL				IRQ	F/F	F/H	F/E
BASE + 6	AD16/12	CNT0	ONE/FH	IROEN	GATE	EXT	PACER	SW

The content of the status register of BASE+6 is the same as that of the control register.

**F/E** FIFO Empty flag

This bit indicates whether the FIFO is empty.

1 means that the FIFO is empty.

**F/H** FIFO Half-full flag

This bit indicates whether the FIFO is half-full.

1 means that the FIFO is half-full.

**F/F** FIFO Full flag

This bit indicates whether the FIFO is full.

1 means that the FIFO is full.

**IRQ** Interrupt flag

This bit indicates the interrupt status.

1 means that an interrupt has occurred.

### C.9 Clear Interrupt and FIFO — BASE+8 and BASE+9

Writing data to either of these two bytes clears the interrupt or the FIFO.

**Table C-9: Register to clear interrupt and FIFO**

Write	Clear Interrupt and FIFO							
Bit #	7	6	5	4	3	2	1	0
BASE + 9	Clear FIFO							
BASE + 8	Clear Interrupt							

### C.10 D/A Output Channel 0 — BASE+10 and BASE+11

The MIC-3716 provides an innovative design as *gate control* for the Analog Output function. It works as a general Analog Output function when you disable the flag (bit 3 (DA0\_LDEN) of BASE+14). That means data will be output immediately. However, when you enable the flag, you need to read these two registers BASE+10 and BASE+11 to output the data to the Analog Output channel.

**Table C-10: Register for load D/A channel 0 data**

Read	Load D/A Channel 0 data							
Bit #	7	6	5	4	3	2	1	0
BASE + 11								
BASE + 10								

### C.11 D/A Output Channel 0 — BASE+10 and BASE+11

The write-only registers of BASE+10 and BASE+11 accept data for the D/A Channel 0 output.

*Table C-11: Register for D/A channel 0 data*

Write	D/A Output Channel 0							
Bit #	7	6	5	4	3	2	1	0
BASE + 11	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
BASE + 10	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

- DA11 ~ DA0** Digital to analog data
- DA0    LSB of the D/A data
- DA15   MSB of the D/A data

### C.12 D/A Output Channel 1 — BASE+12 and BASE+13

The MIC-3716 provides an innovative design as *gate control* for the Analog Output function. It works as a general Analog Output function when you disable the flag (bit 11 (DA1\_LDEN) of BASE+14). That means the data will be output immediately. However, when you enable the flag, you need to read these two registers BASE+12 and BASE+13 to output the data to the Analog Output channel.

**Table C-12: Register for load D/A channel 1 data**

Read	Load D/A Channel 1 data							
Bit #	7	6	5	4	3	2	1	0
BASE + 13								
BASE + 12								

### C.13 D/A Output Channel 1 — BASE+12 and BASE+13

The write-only registers of BASE+12 and BASE+13 accept data for the D/A channel 1 output.

*Table C-13: Register for D/A channel 1 data*

Write	D/A Output Channel 1							
Bit #	7	6	5	4	3	2	1	0
BASE + 13	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
BASE + 12	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

- DA11 ~ DA0** Digital to analog data
- DA0 LSB of the D/A data
- DA15 MSB of the D/A data

### C.14 D/A Reference Control —BASE+14

The write-only register of BASE+14 allows users to set the D/A reference source.

**Table C-14: Register for D/A reference control**

Write	D/A Output Channel 1							
Bit #	7	6	5	4	3	2	1	0
BASE + 15					DA1_LDEN	DA1_I/E	DA0_B/U	DA1_5/10
BASE + 14					DA0_LDEN	DA0_I/E	DA0_B/U	DA0_5/10

**DA<sub>n</sub>\_5/10** The internal reference voltage for the D/A output channel *n*

0 - 5 V

1 - 10 V

**DA<sub>n</sub>\_B/U** for D/A output channel *n*

0 Bipolar

1 Unipolar

**DA<sub>n</sub>\_I/E** Internal or external reference voltage for D/A output channel *n*

0 Internal source

1 External source

**DA<sub>n</sub>\_LDEN** for Gate Control of D/A output channel *n* (Please refer to C.10 and C.12)

0 Disable

1 Enable

### C.15 Digital I/O Registers — BASE+16 and BASE+17

The MIC-3716 offers 16 digital input channels and 16 digital output channels. These I/O channels use the input and output ports at addresses BASE+16 and BASE+17.

**Table C-15: Register for digital input**

Read	Digital Input							
Bit #	7	6	5	4	3	2	1	0
BASE + 17	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
BASE + 16	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

**Table C-16: Register for digital output**

Write	Digital Output							
Bit #	7	6	5	4	3	2	1	0
BASE + 17	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8
BASE + 16	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

---

**Note!**

| The default configuration of the digital output channels is a logic 0.

---

This avoids damaging external devices during system start-up or reset since the power on status is set to the default value.

### C.16 Calibration Registers — BASE+18 and BASE+19

The MIC-3716 offers Calibration registers BASE+16 and BASE+17 for calibrating A/D and D/A.

**Table C-17: Calibration Command and Data Register**

Write	Calibration Command and Data							
Bit #	7	6	5	4	3	2	1	0
BASE + 19					CM3	CM2	CM1	CM0
BASE + 18	D7	D6	D5	D4	D3	D2	D1	D0

**D7 to D0** Calibration data

D0 LSB of the calibration data

D7 MSB of the calibration data

**CM3 to CM0** Calibration Command and *Table C-18* lists the command code for MIC-3716.

**Table C-18: Calibration Command and Data Register**

Meaning	Command Code			
	CM3	CM2	CM1	CM0
A/D bipolar offset adjust	0	0	0	0
A/D unipolar offset adjust	0	0	0	1
PGA offset adjust	0	0	1	0
A/D gain adjust	0	0	1	1
D/A 0 gain 1 adjust (10 V)	0	1	0	0
D/A 0 gain 2 adjust (5 V)	0	1	0	1
D/A 0 bipolar offset adjust	0	1	1	0
D/A 0 unipolar offset adjust	0	1	1	1
D/A 1 gain 1 adjust (10 V)	1	0	0	0
D/A 1 gain 2 adjust (5 V)	1	0	0	1
D/A 1 bipolar offset adjust	1	0	1	0
D/A 1 unipolar offset adjust	1	0	1	1

### C.17 Board ID Registers — BASE+20

The MIC-3716 offers Board ID register BASE+20. With correct Board ID settings, you can easily identify and access each card during hardware configuration and software programming.

**Table C-19: Register for Board ID**

Read	Board ID							
Bit #	7	6	5	4	3	2	1	0
BASE + 20					BD3	BD2	BD1	BD0

### C.18 Programmable Timer/Counter Registers

#### **BASE+24, BASE+26, BASE+28 and BASE+30**

The four registers of BASE+24, BASE+26, BASE+28 and BASE+30 are used for the 82C54 programmable timer/counter. Please refer to *Appendix D. 82C54 Counter Chip Functions* for detailed application information.

---

**Note:**

You have to use a 16-bit (word) command to read/write each register.

---

# Appendix D. 82C54 Counter Chip

## Function

---

### D.1 The Intel 82C54

The MIC-3716 uses one Intel 82C54 compatible programmable interval timer/counter chip. The popular 82C54 offers three independent 16-bit counters, counter 0, counter 1 and counter 2. Each counter has a clock input, control gate and an output. You can program each counter for maximum count values from 2 to 65535.

The 82C54 has a maximum input clock frequency of 10 MHz. The MIC-3716 provides 10 MHz input frequencies to the counter chip from an on-board crystal oscillator.

#### Counter 0

On the MIC-3716, counter 0 can be a 16-bit timer or an event counter, selectable by users. When the clock source is set as an internal source, counter 0 is a 16-bit timer; when set as an external source, then counter 0 is an event counter and the clock source comes from CNT0\_CLK. The counter is controlled by CNT0\_GATE. When CNT0\_GATE input is high, counter 0 will begin to count.

#### Counter 1 & 2

Counter 1 and counter 2 of the counter chip are cascaded to create a

32-bit timer for the pacer trigger. A low-to-high edge of counter 2 output (PACER\_OUT) will trigger an A/D conversion. At the same time, you can use this signal as a synchronous signal for other applications.

## D.2 Counter Read/Write and Control Registers

The 82C54 programmable interval timer uses four registers at addresses  $\text{BASE} + 24(\text{Dec})$ ,  $\text{BASE} + 26(\text{Dec})$ ,  $\text{BASE} + 28(\text{Dec})$  and  $\text{BASE} + 30(\text{Dec})$  for read, write and control of counter functions.

Register functions appear below:

Register	Function
$\text{BASE} + 24(\text{Dec})$	Counter 0 read/write
$\text{BASE} + 26(\text{Dec})$	Counter 1 read/write
$\text{BASE} + 28(\text{Dec})$	Counter 2 read/write
$\text{BASE} + 30(\text{Dec})$	Counter control word

Since the 82C54 counter uses a 16-bit structure, each section of read/write data is split into a least significant byte (LSB) and most significant byte (MSB). To avoid errors, it is important that you make read/write operations in pairs and keep track of the byte order.

The data format for the control register is as below:

<b>BASE+30(Dec) 82C54 control, standard mode</b>								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	SC1	SC0	RW1	RW0	M2	M1	M0	BCD

## Description

### SC1 & SC0 Select counter

Counter	SC1	SC0
0	0	0
1	0	1
2	1	0
Read-back command	1	1

### RW1 & RW0 Select read / write operation

Operation	RW1	RW0
Counter latch	0	0
Read/write LSB	0	1
Read/write MSB	1	0
Read/write LSB first, then MSB	1	1

### M2, M1 & M0 Select operating mode

M2	M1	M0	Mode	Description
0	0	0	0	Stop on terminal count
0	0	1	1	Programmable one shot
X	1	0	2	Rate generator
X	1	1	3	Square wave rate generator
1	0	0	4	Software triggered strobe
1	0	1	5	Hardware triggered strobe

### BCD Select binary or BCD counting

BCD	Type
0	Binary counting 16-bits
1	Binary coded decimal (BCD) counting

If you set the module for binary counting, the count can be any number from 0 up to 65535. If you set it for BCD (Binary Coded Decimal) counting, the count can be any number from 0 to 9999.

If you set both SC1 and SC0 bits to 1, the counter control register is in read-back command mode. The control register data format then becomes:

BASE + 30(Dec) 82C54 control, read-back mode								
Bit	D7	D6	D5	D4	D3	D2	D1	D0
Value	1	1	CNT	STA	C2	C1	C0	X

**CNT = 0** Latch count of selected counter(s)

**STA = 0** Latch status of selected counter(s)

**C2, C1 & C0** Select counter for a read-back operation

C2 = 1 select Counter 2

C1 = 1 select Counter 1

C0 = 1 select Counter 0

If you set both SC1 and SC0 to 1 and STA to 0, the register selected by C2 to C0 contains a byte which shows the status of the counter.

The data format of the counter read/write register then becomes:

---

<b>BASE+24/26/28(Dec) Status read-back mode</b>								
<b>Bit</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>Value</b>	OUT	NC	RW1	RW0	M2	M1	M0	BCD

---

**OUT** Current state of counter output

**NC** Null count is 1 when the last count written to the counter register has been loaded into the counting element

## **D.3 Counter Operating Modes**

### **MODE 0 – Stop on Terminal Count**

The output will initially be low after you set this mode of operation. After you load the count into the selected count register, the output will remain low and the counter will count. When the counter reaches the terminal count, its output will go high and remain high until you reload it with the mode or a new count value. The counter continues to decrement after it reaches the terminal count. Rewriting a counter register during counting has the following results:

1. Writing to the first byte stops the current counting.
2. Writing to the second byte starts the new count.

### **MODE 1 – Programmable One-shot Pulse**

The output is initially high. The output will go low on the count following the rising edge of the gate input. It will then go high on the terminal count. If you load a new count value while the output is low, the new value will not affect the duration of the one-shot pulse until the succeeding trigger. You can read the current count at any time without affecting the one-shot pulse. The one-shot is re-triggerable, thus the output will remain low for the full count after any rising edge at the gate input.

### **MODE 2 – Rate Generator**

The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the counter register. If you reload the counter register between output

pulses, the present period will not be affected, but the subsequent period will reflect the value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. You can thus use the gate input to synchronize the counter.

With this mode the output will remain high until you load the count register. You can also synchronize the output by software.

### **MODE 3 – Square Wave Generator**

This mode is similar to Mode 2, except that the output will remain high until one half of the count has been completed (for even numbers), and will go low for the other half of the count. This is accomplished by decreasing the counter by two on the falling edge of each clock pulse. When the counter reaches the terminal count, the state of the output is changed, the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the count by 2. After time-out, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by two until time-out, then the whole process is repeated. In this way, if the count is odd, the output will be high for

$(N+1)/2$  counts and low for  $(N-1)/2$  counts.

#### **MODE 4 –Software-Triggered Strobe**

After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period then go high again. If you reload the count register during counting, the new count will be loaded on the next CLK pulse. The count will be inhibited while the GATE input is low.

#### **MODE 5 – Hardware-Triggered Strobe**

The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is re-triggerable.

## D.4 Counter Operations

### Read/Write Operation

Before you write the initial count to each counter, you must first specify the read/write operation type, operating mode and counter type in the control byte and write the control byte to the control register [BASE + 30(Dec)].

Since the control byte register and all three counter read/write registers have separate addresses and each control byte specifies the counter it applies to (by SC1 and SC0), no instructions on the operating sequence are required. Any programming sequence following the 82C54 convention is acceptable.

There are three types of counter operations: Read/load LSB, read /load MSB and read /load LSB followed by MSB. It is important that you make your read/write operations in pairs and keep track of the byte order.

### Counter Read-back Command

The 82C54 counter read-back command lets you check the count value, programmed mode and current states of the OUT pin and Null Count flag of the selected counter(s). You write this command to the control word register. The format is as shown at the beginning of this section. The read-back command can latch multiple counter output latches. Simply set the CNT bit to 0 and select the desired counter(s). This single command is functionally equivalent to multiple counter

latch commands, one for each counter latched.

The read-back command can also latch status information for selected counter(s) by setting STA bit = 0. The status must be latched to be read; the status of a counter is accessed by a read from that counter. The counter status format appears at the beginning of the chapter.

## **Counter Latch Operation**

Users often want to read the value of a counter without disturbing the count in progress. You do this by latching the count value for the specific counter, and then read the value.

The 82C54 supports the counter latch operation in two ways. The first way is to set bits RW1 and RW0 to 0. This latches the count of the selected counter in a 16-bit hold register. The second way is to perform a latch operation under the read-back command. Set bits SC1 and SC0 to 1 and CNT = 0. The second method has the advantage of operating several counters at the same time. A subsequent read operation on the selected counter will retrieve the latched value.

*(This page is left blank for hard printing.)*

# Appendix E. Manual Calibration

---

## E.1 A/D Calibration

Regular and proper calibration procedures ensure the maximum possible accuracy. It is easy to complete the A/D calibration procedure automatically (i.e. through software calibration) by executing the A/D calibration program *AutoCali.EXE*. Therefore, it is not necessary to adjust the hardware settings of the MIC-3716. However, the following calibration steps are also provided for your reference in case manual calibration is needed:

1. Adjust the on board reference voltage. First, adjust VR1 until the reference voltage on TP4 has reached +5.0000 V. Next, to write 0x0080, 0x0180, 0x0280 and 0x0380 sequentially to the *Calibration Command and Data register* (BASE+18) set MIC-3716 to AI software trigger and calibration mode.
2. Adjust the **PGA** offset voltage. First, write any value to BASE+9 to clear FIFO. Then set A/D channel to channel 0.
3. Write the *value* from 0x0200 to 0x02FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times then average those data for each *value*. After that, compare the average data of the range between  $\pm 5$  V and  $\pm 0.625$  V, to see whether the discrepancy is less than 2 LSB. If so, go to the next step. Otherwise, you must change the *value* and repeat the whole procedure in this step again until the

discrepancy is less than 2 LSB.

4. Adjust the **BIPOLAR** offset voltage. First, write any value to BASE+9 to clear FIFO. Then set the A/D channel to channel 0, and to set the range as -5 V to +5 V.
5. Write the *value* from 0x0000 to 0x00FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times, then average those data for each *value*. Check if the average data is close to 32767.5. If so, go to the next step. Otherwise, you must change the *value* and repeat the whole procedure in this step again until the average data is close to 32767.5.
6. Adjust **UNIPOLAR** offset voltage. First, write any value to BASE+9 to clear FIFO. Then set A/D channel to channel 0, and set the range as 0 V to 10 V.
7. Write the *value* from 0x0100 to 0x01FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times, then average those data for each *value*. See whether the average data is close to 32767.5. If so, go to the next step. Otherwise, you must change the *value* and repeat all the procedure in this step again until the average data is close to 32767.5.
8. Adjust **GAIN** offset voltage. First, write any value to BASE+9 to clear FIFO. Then set A/D channel to channel 2, and set the range as -5 V to +5 V.

9. Write the *value* from 0x0300 to 0x03FF sequentially to the *Calibration Command and Data register (BASE+18)*, and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times then average those data for each *value*. See whether the average data is close to 65534.6. If so, go to the next step. Otherwise, you must change the *value* and repeat the whole procedure in this step again until the average data is close to 65534.6.
10. Repeat steps 2 to 9 several times.

**Table E-1: A/D binary code table**

A/D code		Mapping Voltage	
Hex.	Dec.	Bipolar	Unipolar
0000h	0	-FS	0
7FFFh	32767	-1 LSB	0.5 FS - 1 LSB
8000h	32768	0	0.5 FS
FFFFh	65535	+FS - 1 LSB	FS - 1 LSB

---

**Note**

- $1 \text{ LSB} = \text{FS} / 65535$  for Unipolar  
(For example:  $1\text{LSB} = 10 / 65535$ , while the range is 0 V to 10 V)
  - $1 \text{ LSB} = +\text{FS} / 32768$  for Bipolar  
(For example:  $1\text{LSB} = 5 / 32768$ , while the range is -5 V to +5 V)
-

## E.2 D/A Calibration

You can select an onboard +5V or +10V *internal reference voltage* or an *external voltage* as your analog output reference voltage. If you use an external reference, connect the reference voltage within the  $\pm 10\text{V}$  range to the reference input of the D/A output channel you want to calibrate. Then adjust the gain value, unipolar offset voltage, bipolar offset voltage, respectively, of D/A channels 0 and 1 with the *Calibration Command and Data register* (BASE+18).

---

### Note:

Using a precision voltmeter to calibrate the D/A outputs is recommended.

---

The auto-calibration D program *AutoCali.EXE* helps you finish the D/A calibration procedure automatically. In order to get the maximum possible accuracy of the D/A channels, you need to calibrate the A/D channels first. Although the procedure is not necessary, the following calibration steps are provided below for your reference in case you want to implement the calibration yourself:

1. Write 0x0400, 0x0500, 0x0600, 0x0700, 0x0800, 0x0900, 0x0A00 and 0x0B00 sequentially to the *Calibration Command and Data register* (BASE+18). Next, set the MIC-3716 to AI software trigger and calibration mode. After that, set the A/D channel to the corresponding D/A channel. That means, connect A/D channel 4 to D/A channel 0, and connect A/D channel 6 to D/A channel 1.
2. Adjust **GAIN 10V** calibration. First, write any value to BASE+9 to clear FIFO. Then set the A/D range as 0 V to 10 V, and set the

- D/A range as 0 V to 10 V. Next, write 0xFFFF to the corresponding *D/A registers* (BASE+10 and BASE+12).
3. Write the **value** from 0x0400 to 0x04FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times to average those data for each **value**. After that, see whether the average data is close to 65534.6. If so, go to the next step. Otherwise, you must change the **value** and repeat the whole procedure in this step again until the average data is close to 65534.6.
  4. Adjust **GAIN 5V** calibration. First, write any value to BASE+9 to clear FIFO. Then set the A/D range as -5 V to +5 V, and set the D/A range as 0 V to 5 V. Next, write 0xFFFF to the corresponding *D/A registers* (BASE+10 and BASE+12).
  5. Write the **value** from 0x0500 to 0x05FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times to average those data for each **value**. After that, see whether the average data is close to 65534.6. If so, go to the next step. Otherwise, you must change the **value** and repeat the whole procedure in this step again until the average data is close to 65534.6.
  6. Adjust **BIPOLAR** offset calibration. First, write any value to BASE+9 to clear FIFO. Then set the A/D range as -5 V to +5 V, and set the D/A range as -5 V to +5 V. Next, write 0x0000 to the corresponding *D/A registers* (BASE+10 and BASE+12).

7. Writing the *value* from 0x0600 to 0x06FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times to average those data for each *value*. After that, see whether the discrepancy is less than 0.4 LSB. If so, go to the next step. Otherwise, you must change the *value* and repeat the whole procedure in this step again until the discrepancy is less than 0.4 LSB.
8. Adjust **UNIPOLAR** offset calibration. First, write any value to BASE+9 to clear FIFO. Then set the A/D range as -5 V to +5 V, and set the D/A range as 0 V to 5 V. Next, write 0x8000 to the corresponding *D/A registers* (BASE+10 and BASE+12).
9. Write the *value* from 0x0600 to 0x06FF sequentially to the *Calibration Command and Data register* (BASE+18), and get each bipolar range's data by the software trigger A/D method. Repeat this procedure 1000 times then average those data for each *value*. After that, see whether the average data is close to 32767.5. If so, go to the next step. Otherwise, you must change the *value* and repeat the whole procedure in this step again until the average data close is to 32767.5.
10. Repeat steps 2 to 9 several times.

**Table E-2: D/A binary code table**

A/D code		Mapping Voltage	
Hex.	Dec.	Bipolar	Unipolar
0000h	0	-FS	0
7FFFh	32767	-1 LSB	0.5 FS – 1 LSB
8000h	32768	0	0.5 FS
FFFFh	65535	+FS – 1 LSB	FS – 1 LSB

---

**Note**

- $1 \text{ LSB} = \text{FS} / 65535$  for Unipolar  
(For example:  $1\text{LSB} = 10 / 65535$ , while the range is 0 V to 10 V)
  - $1 \text{ LSB} = +\text{FS} / 32768$  for Bipolar  
(For example:  $1\text{LSB} = 5 / 32768$ , while the range is –5 V to +5 V)
-