INDUSTRIAL COMPUTER SOURCE®

Model AOB8/16 & AOB16/16
Product Manual

MANUAL NUMBER : 00650-010-6B

INDUSTRIAL COMPUTER SOURCE® http:/www.indcompsrc.com

6260 SEQUENCE DRIVE, SAN DIEGO, CA 92121-4371 (619) 677-0877 (FAX) 619-677-0895

INDUSTRIAL COMPUTER SOURCE EUROPE *TEL 01.69.18.74.40 FAX 01.64.46.40.42* • ***INDUSTRIAL COMPUTER SOURCE (UK) LTD*** *TEL 01243-523500 FAX 01243-532949*

FOREWARD

This product manual provides information to install, operate and or program the referenced product(s) manufactured or distributed by Industrial Computer Source. The following pages contain information regarding the warranty and repair policies.

Technical assistance is available at: **1-800-480-0044**.

Manual Errors, Omissions and Bugs: A "Bug Sheet" is included as the last page of this manual. Please use the "Bug Sheet" if you experience any problems with the manual that requires correction.

The information in this document is provided for *reference* only. Industrial Computer Source does not assume any liability arising out of the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Industrial Computer Source, nor the rights of others.

Guarantee

A thirty day money-back guarantee is provided on all **standard** products sold. **Special order products** are covered by our Limited Warranty, *<u>however they may not be returned for refund or credit</u>*. *EPROMs, RAM, Flash EPROMs or other forms of solid electronic media are not returnable for credit* **-** *but for replacement only. Extended Warranty available. Consult factory.*

Refunds

In order to receive refund on a product purchase price, the product must not have been damaged by the customer or by the common carrier chosen by the customer to return the goods, and the product must be returned complete (meaning all manuals, software, cables, etc.) within 30 days of receipt and in as-new and resalable condition. The **Return Procedure** must be followed to assure prompt refund.

Restocking Charges

Product returned *after* 30 days, and *before* 90 days, of the purchase will be subject to a **minimum** 20% restocking charge and any charges for damaged or missing parts.

Products not returned within 90 days of purchase, or products which are not in as-new and re-saleable condition, are not eligible for credit return and will be returned to the customer.

Limited Warranty

One year limited warranty on all products sold with the exception of the “Performance Series” I/O products, which are warranted to the original purchaser, for as long as they own the product, subject to all other conditions below, including those regarding neglect, misuse and acts of God. Within one year of purchase, Industrial Computer Source will repair or replace, at our option, any defective product. At any time after one year, we will repair or replace, at our option, any defective “Performance Series” I/O product sold. This does not include products damaged in shipment, or damaged through customer neglect or misuse. Industrial Computer Source will service the warranty for all standard catalog products for the first year from the date of shipment. After the first year, for products not manufactured by Industrial Computer Source, the remainder of the manufacturer's warranty, if any, will be serviced by the manufacturer directly.

The **Return Procedure** must be followed to assure repair or replacement. Industrial Computer Source will normally return your replacement or repaired item via UPS Blue. *Overnight delivery or delivery via other carriers is available at additional charge.*

The limited warranty is void if the product has been subjected to alteration, neglect, misuse, or abuse; if any repairs have been attempted by anyone other than Industrial Computer Source or its authorized agent; or if the failure is caused by accident, acts of God, or other causes beyond the control of Industrial Computer Source or the manufacturer. Neglect, misuse, and abuse shall include any installation, operation, or maintenance of the product other than in accordance with the owners' manual.

No agent, dealer, distributor, service company, or other party is authorized to change, modify, or extend the terms of this Limited Warranty in any manner whatsoever. Industrial Computer Source reserves the right to make changes or improvements in any product without incurring any obligation to similarly alter products previously purchased.

®

Shipments not in compliance with this Guarantee and Limited Warranty Return Policy will not be accepted by Industrial Computer Source.

Return Procedure

For any Limited Warranty or Guarantee return, please contact Industrial Computer Source's Customer Service at **1-800-480-0044** and obtain a Return Material Authorization (RMA) Number. All product(s) returned to Industrial Computer Source for service or credit **must** be accompanied by a Return Material Authorization (RMA) Number. Freight on all returned items **must** be prepaid by the customer who is responsible for any loss or damage caused by common carrier in transit. Returns for Warranty **must** include a Failure Report for each unit, by serial number(s), as well as a copy of the original invoice showing date of purchase.

> To reduce risk of damage, returns of product must be in an Industrial Computer Source shipping container. If the original container has been lost or damaged, new shipping containers may be obtained from Industrial Computer Source Customer Service at a nominal cost.

Limitation of Liability

In no event shall Industrial Computer Source be liable for any defect in hardware or software or loss or inadequacy of data of any kind, or for any direct, indirect, incidental, or consequential damages in connection with or arising out of the performance or use of any product furnished hereunder. Industrial Computer Source liability shall in no event exceed the purchase price of the product purchased hereunder. The foregoing limitation of liability shall be equally applicable to any service provided by Industrial Computer Source or its authorized agent.

Some *Sales Items* and *Customized Systems* are **not** subject to the guarantee and limited warranty. However in these instances , any deviations will be disclosed prior to sales and noted in the original invoice. ***Industrial Computer Source reserves the right to refuse returns or credits on software or special order items.***

Advisories

Three types of advisories are used throughout the manual to stress important points or warn of potential hazards to the user or the system. They are the Note, the Caution, and the Warning. Following is an example of each type of advisory:

Note: The note is used to present special instruction, or to provide extra information which may help to simplify the use of the product.

CAUTION!

A Caution is used to alert you to a situation which if ignored may cause injury or damage equipment.

WARNING!

A Warning is used to alert you of a situation which if ignored will cause serious injury.

Cautions and Warnings are accented with triangular symbols. The exclamation symbol is used in all cautions and warnings to help alert you to the important instructions. The lightning flash symbol is used on the left hand side of a caution or a warning if the advisory relates to the presence of voltage which may be of sufficient magnitude to cause electrical shock.

Use caution when servicing any electrical component. We have tried to identify the areas which may pose a Caution or Warning condition in this manual; however, Industrial Computer Source does not claim to have covered all situations which might require the use of a Caution or Warning.

You must refer to the documentation for any component you install into a computer system to insure proper precautions and procedures are followed.

Table of Contents

List of Figures

Current Revision 6B

August 1997

Chapter 1: Installation

Backing up the Disk

The software provided with the AOBx/16 is in DOS, diskette format. You should make back-up copies for everyday use and place your original master diskette in a safe location. You may make as many back-up copies as needed.

The easiest way to make a back-up copy is to use the DOS DISKCOPY utility. First, use the DOS command FORMAT/S to format a new diskette and then copy files from the master diskette.

In a single-drive system, the command is: DISKCOPY A: A:

You will need to swap disks as requested by the system.

In a two-disk system, the command is: DISKCOPY A: B:

This will copy the contents of the master disk in drive A to the back-up disk in drive B.

Hard Disk Installation

The files contained on the master diskette may also be copied onto your hard disk. To do this perform the following:

1. Place the master diskette into a floppy drive.
2. Change the active drive to the drive that has the master diskette installed. For example, if the diskette is the A drive, type A:
3. Type INSTALL and follow the screen prompts.

Files contained on the disk are stored in separate directories as follows:

ROOT DIRECTORY:	Contains the FINDBASE PROGRAM that will help you to decide what base address to use with the card. Also contains the DA16SET.EXE setup and calibration program.
PSAMPLES:	Contains Pascal samples and the Pascal-linkable driver.
CSAMPLES:	Contains “C” samples and the “C”-linkable driver.
BSAMPLES:	Contains the BASIC and QuickBASIC samples as well as the binary and linkable drivers.
VBACCESS:	VisualBASIC utility driver that includes PEEK and POKE statements for reading and writing RAM as well as INPORT and OUTPORT for reading and writing I/O. The driver is in the form of a DLL and allows you to access hardware as if the language was designed for it when you use VisualBASIC for Windows.

Installing the Card

Before installing the card carefully read the Address Selection and Option Selection sections of this manual and configure the card according to your requirements. Be especially careful with address selection. If the addresses of two installed functions overlap you will experience unpredictable computer behavior. If unsure what locations are available, you can use the FINDBASE program provided on our diskette to locate blocks of available addresses.

To install the card:

1. Turn off computer power.
2. Remove the computer cover.
3. Remove the blank I/O backplate.
4. Set switches for selected options. See the option selection section of this manual.
5. Select the base address on the card. See the address selection section of this manual
6. Install the card in an I/O expansion slot.
7. Install the I/O cable.
8. Inspect for proper fit of the card and cables, tighten screws.
9. Replace the computer cover and apply power.

Chapter 2: Functional Description

The AOB8/16 and AOB16/16 are full-size cards that can be installed in any long I/O slot of PC-AT class computers. They contain either 8 or 16 double-buffered digital-to-analog converters (DAC) and provide 8 or 16 independent analog output channels of 16-bit resolution. Each analog output channel can be configured for ranges of:

0V to +5V

0V to +10V

-5V to +5V

-10V to +10V

The Option Selection section of this manual contains a description of how to make these selections.

The analog output channels have a double-buffered input for single-step update and each is addressed at its own I/O location. The DACs have a two-byte (8LSB's+8MSB's) loading structure. The analog outputs can be updated either independently or simultaneously.

Finally, the AOBx/16 contain automatic reset circuits which reset both D/A outputs to all zeroes at system power-on. Upon power-up or hardware reset, the DAC registers are initialized to a "zero" value and the card is set in the Simultaneous Update mode. The cards also support a unique "Software Clear" capability. This feature permits resetting the DAC output to zero volts without changing the output mode.

Software provided with the card includes setup and calibration programs and some sample programs. The setup and calibration program provides pictorial representation and menu selection on the computer monitor. For setup, of course, it is not necessary that the card be plugged into the computer.

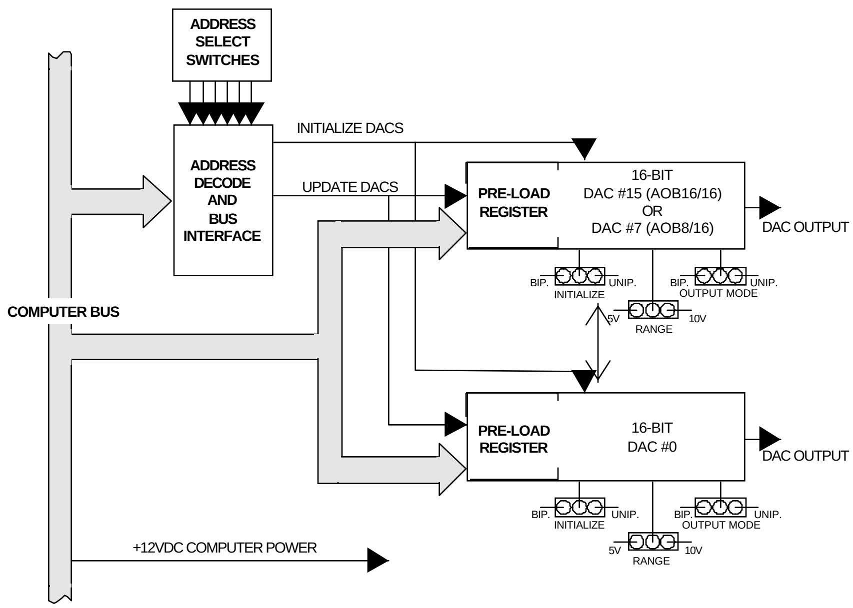


Figure 2-1: Block Diagram

Chapter 3: Option Selection

Voltage output ranges are determined by jumper placement as described in the following paragraphs. Also, the method to update D/A outputs is programmable as described here and in the Programming section of this manual.

Output Ranges

To select output voltage ranges (either unipolar or bipolar) set three jumpers located below each DAC output chip. The jumpers select the polarity and range of each DAC channel.

Voltage Range	Mode "M" and Initialize Jumper "I"	Range Jumper "R"
0 to +5 V	Unipolar	5V
-5 to +5 V	Bipolar	5V
0 to +10 V	Unipolar	10 V
-10 to +10 V	Bipolar	10 V

Analog Outputs Update

Analog outputs are updated under program control in either one of two ways:

(a) Each channel is normally updated individually when new data are written to the related high-byte base address. This "individual update" mode may be set by a special read operation as defined in the programming chapter.

OR

(b) The outputs of all D/A's may be updated <u>simultaneously</u>. This is done by first enabling simultaneous updating for all outputs and then preloading the high and low bytes of each DAC and then initiating a simultaneous update. (Simultaneous update mode is the default on power up.)

Refer to the Programming section of this manual for more detail.

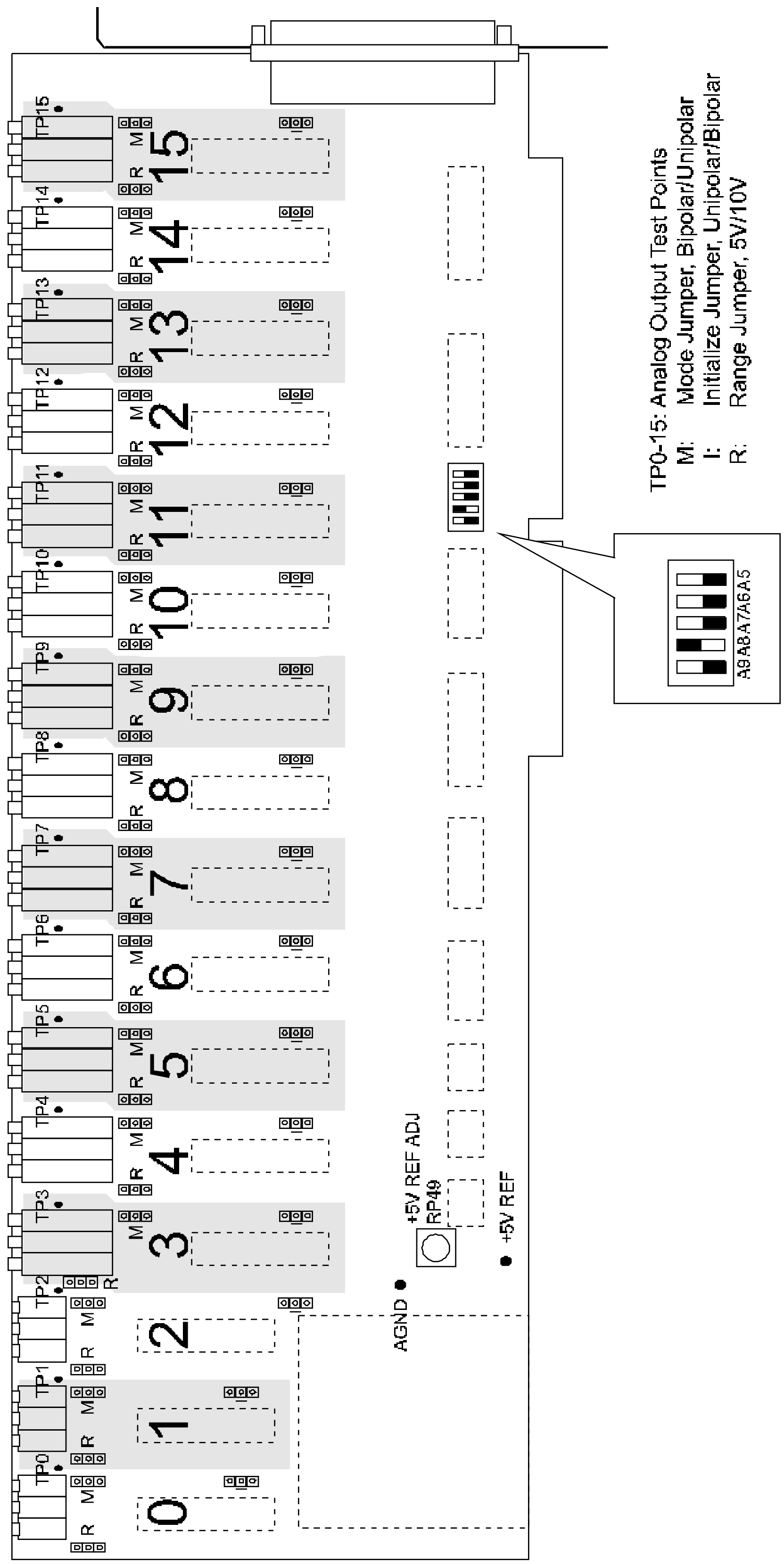


Figure 3-1: Option Selection Map

Chapter 4: Address Selection

The AOB8/16 and AOB16/16 require 16 or 32 consecutive address locations in I/O space, respectively. The starting, or base address, can be selected anywhere within an I/O address range 100-3FF hex (except 1F0 through 1F8) in AT-class computers and 200- 3FF in XT-class computers, providing that the address does not overlap with other functions. If in doubt refer to the table below for a list of standard address assignments. The Base Address Locator program FINDBASE provided on diskette will assist you in selecting a base address that will avoid this conflict.

Hex Range	Usage
000-0FF	Internal System - Not Usable
1F0-1FF	AT Hard Disk
200-207	Game Control
278-27F	Parallel Port (LPT2)
238-23B	Bus Mouse
2E8-2EF	Asynchronous Communications (COM4)
2F8-2FF	Asynchronous Communications (COM2)
300-31F	Prototype Card
320-32F	XT Hard Disk
378-37F	Parallel Port (LPT1)
380-38F	SDLC Communications
3A0-3AF	SDLC Communications
3B0-3BB	MDA
3BC-3BF	Alt. Parrallel Port
3C0-3CF	EGA
3DO-3DF	CGA
3E8-3EF	Asynchronous Communications (COM3)
3F0-3F7	Floppy Disk
3F8-3FF	Asynchronous Communications (COM1)

Figure 4-1: Standard Address Assignments for PC and PC/XT Computers

The AOBx/16 base address bits A5 through A9 are set by DIP switch S1. The setup program provided on diskette with your card includes an interactive base-address selection program. The computer monitor presents a pictorial display of the DIP switch and, when you enter your desired hex base address, the display changes to show proper switch settings for that address.

To understand how this works, consider the following. In order to select the base address, convert the desired address to binary form. Then *for each "1" of binary address set the corresponding DIP switch to OFF, and for each "0" of binary address set the corresponding switch to ON.*

Here's an example showing how to program the base address to hex 300:

1. Convert hex 300 to binary

   300 (hex) = 11 0000 0000 (binary)

2. Set the Address Selection DIP Switches

The AOBx/16 card occupy 16 and 32 bytes of I/O address space, respectively. Address lines A5 through A9 are used to select the base address via DIP switches marked with the same names. Address lines A0 - A4 are used to address registers at the digital-to-analog converters and there are no DIP switches for these five lines.

Address	1	1	0	0	0	0	0	0	0	0
Switch	A9	A8	A7	A6	A5	None				
Setting	OFF	OFF	ON	ON	ON	None				

Chapter 5: Programming

Programming the AOBx/16 is very straightforward as there are only two operating modes, three sets of jumpers, and one unique addition. The basic operation of a Digital to Analog card is to write a 16-bit value to a Digital to Analog Converter (DAC) preload register where it is buffered and loaded with an update command to a DAC register which produces the corresponding analog output (Defined by the range and polarity jumpers for that channel).

Upon power-up, or hardware reset, the DAC registers are initialized to a “zero” value and the card is set in Simultaneous Update mode. This ensures that upon power-up the outputs start at zero volts out. (**Note:** The “I” initialize and “M” polarity mode jumpers should be set identically or the DAC register will be initialized to the incorrect value.) Since the preload register is not cleared upon power-up, but left at an undefined value, a known value must be written to the preload registers before using an update command.

Simultaneous Update Mode is the power-up or default mode of operation for the DAC card. When a value is written to a DAC address the output does not change until an output update is commanded via a read to the BASE+8 address. (Alternatively, a read to BASE+10 will update the DAC registers and switch the board to Automatic Update Mode.) While in Simultaneous Update Mode, a single read will load all DAC registers with the value waiting in the preload registers causing all outputs to be updated and changed simultaneously.

Automatic Update Mode is the power-up default configuration which changes the DAC output immediately after the new value high-byte is written to the DAC address. If the card is in Simultaneous Update Mode a read to BASE+2 address will change the card back to Automatic Update Mode without updating the outputs. Or, a read to BASE+10 will update all outputs simultaneously and then release the card to the Automatic Update Mode.

Software Clear is a unique addition to our DAC card which resets the DAC similar to a hardware reset without changing the operating mode. Just as a hardware reset, the zero output depends on the proper setup of the initialize and polarity mode jumpers (See the power-up paragraph) to produce a zero output. Since the preload registers are not cleared the previous output will be restored from the preload register when the appropriate update command is issued to the DAC channel.

The AOB8/16 and AOB16/16 cards use 16 and 32 consecutive I/O addresses, respectively. The I/O address map is as follows:

Address	Write *	Read
Base + 0	DAC 0 Low Byte	Place card in Simultaneous Mode without updating outputs.
Base + 1	DAC 0 High Byte	
Base + 2	DAC 1 Low Byte	Release card from Simultaneous Mode without updating outputs
Base + 3	DAC 1 High Byte	
Base + 4	DAC 2 Low Byte	
Base + 5	DAC 2 High Byte	
Base + 6	DAC 3 Low Byte	
Base + 7	DAC 3 High Byte	
Base + 8	DAC 4 Low Byte	Update all outputs and place card in Simultaneous Mode
Base + 9	DAC 4 High Byte	
Base + 10	DAC 5 Low Byte	Update all outputs and release card from Simultaneous Mode
Base + 11	DAC 5 High Byte	
Base + 12	DAC 6 Low Byte	
Base + 13	DAC 6 High Byte	
Base + 14	DAC 7 Low Byte	Set all outputs to zero
Base + 15	DAC 7 High Byte	Release zero latch
Base + 16	DAC 8 Low Byte	
Base + 17	DAC 8 High Byte	
Base + 18	DAC 9 Low Byte	
Base + 19	DAC 9 High Byte	
Base + 20	DAC 10 Low Byte	
Base + 21	DAC 10 High Byte	
Base + 22	DAC 11 Low Byte	
Base + 23	DAC 11 High Byte	
Base + 24	DAC 12 Low Byte	
Base + 25	DAC 12 High Byte	
Base + 26	DAC 13 Low Byte	
Base + 27	DAC 13 High Byte	
Base + 28	DAC 14 Low Byte	
Base + 29	DAC 14 High Byte	
Base + 30	DAC 15 Low Byte	
Base + 31	DAC 15 High Byte	

* Although it is possible to write the low and high bytes separately as shown above, it is much easier to write both bytes with a single OUT DX, AX instruction. In that case, only even addresses are written.

BIT	D7	D6	D5	D4	D3	D2	D1	D0
Low Byte	B7	B6	B5	B4	B3	B2	B1	B0
High Byte	B15	B14	B13	B12	B11	B10	B9	B8

Figure 5-1: Data Format

For UNIPOLAR ranges: For Unipolar ranges, data are in true binary form.

0000 0000 0000 0000 = ZERO
1000 0000 0000 0000 = 1/2 SCALE
1111 1111 1111 1111 = FULL SCALE
| |
MSB or B1<— | | —> B16 or LSB

For BIPOLAR ranges: For Bipolar ranges, data are in complementary offset binary form.

0000 0000 0000 0000 = + FULL SCALE
1000 0000 0000 0000 = ZERO
1111 1111 1111 1111 = - FULL SCALE
| |
MSB or B1 <——| |——> B16 or LSB

This page intentionally left blank

Chapter 6: Software

The AOBx/16 cards are straightforward to program. The following example is in BASIC, but for languages such as C or Pascal the procedure is simplified by their support of two-byte output operations:

To output an analog value with 16-bit resolution, a corresponding decimal number N between 0 and 65536 is calculated (2^12 = 65536).

N/65536 = V(out)/V(full scale)

Then the number is split between high and low bytes, as follows:

```
H% = INT (N / 256)
L% = N - (H% * 256)
```

Next the data are written to the selected analog output channel. (See the preceding I/O Address Map.) In this example, we will assume analog output on channel zero (AO 0).

```
OUT (BASE + 0), L%
OUT (BASE + 1), H%
```

For simplicity, it was assumed that the simultaneous-update capability was not used.

Examples of this routine are found on the sample disk along with examples in other languages.

VisualBASIC™ Utility Driver

Extensions to the Standard Edition VisualBASIC language are provided on the diskette provided with your card. The extensions are in a directory named VBACCES. These extensions are in the form of a .DLL, a .GBL, and a VisualBASIC sample. Together these files allow you to access the port and main memory space in a fashion similar to BASIC, QuickBASIC, Pascal, C/C^{++}, Assembly, and most other standard languages.

To use these files in a VisualBASIC program, you must create a .MAK file (File l New Project) similar to the sample provided (or else, modify your existing project file) and include the .GBL file (File | Add File). Once this has been done, VisualBASIC will be enhanced with the addition of the following functions.

InPortb

Function: Reads a byte from a hardware port. Due to limitations of VisualBASIC, the number is returned in an integer.

Declaration: `function InPortb(byval address as integer) as integer`

InPort

Function: Reads an integer from a hardware port. This function returns the 16-bit value obtained from reading the low byte from *address* and the high byte from *address+1*.

Declaration: `function InPort(byval address as integer) as integer`

OutPortb

Function: Writes the lower eight bits of *value* to the hardware port at *address*. This function returns the value output.

Declaration: `function OutPortb(byval address as integer, byval value as integer) as integer`

OutPort

Function: Writes all 16 bits of *value* to the hardware port at *address*. This function returns the value output.

Declaration: `function OutPort(byval address as integer, byval value as integer) as integer`

Peek

Function: Reads a byte from main memory (DRAM).

Declaration: `function Peek(byval segment as integer, byval offset as integer) as integer`

Poke

Function: Writes the lower eight bits of *value* to *segment*:*offset.*

Declaration: `function Poke(byval segment as integer, byval offset as integer, byval value as integer) as integer`

Note: That in all of the above functions, an inherent limitation of BASIC in general, and VisualBASIC in particular, makes the values sent less intuitive. All integers in BASIC are signed numbers, wherein data are stored in two's complement form. All bit patterns must be converted to-and-from this two's complement form if meaningful display is required. Otherwise, values returned from the InPortb function will be -128 to 127, rather than 0 to 255. An alternative is to perform all assignments in hexadecimal, rather then decimal form.

Before the program will execute, the .GBL file must be modified to include the path to the VBACCES.DLL as appropriate for your system. Merely replace the statement "VBACCES.DLL" with "*drive:path*\VBACCES.DLL".

As an alternative to changing the source code, you can copy the VBACCES.DLL file into your Windows directory. This will allow multiple programs to find the same .DLL without having to know where it is located. Just leave off all references to a path in the .GBL file as shown in the sample.

This page intentionally left blank

Chapter 7: Calibration

Periodic calibration of the AOBx/16 cards are recommended if it is used in extreme environmental conditions. The card uses very stable components but vibration, or high-low temperature cycles might result in slight analog output errors.

Factory calibration and periodic calibration of the card includes adjustment of the internal reference voltage unless you are using an external reference voltage.

The suggested sequence for calibration is:

a. Set base address for the card
b. Set range and polarity for each channel
c. Adjust 5V Reference Voltage
d. Adjust Unipolar zero on each channel
e. Adjust Unipolar full scale of each channel
f. Adjust Bipolar zeroes of each channel
g. Check Bipolar negative full scale of each channel
h. Check Bipolar zero of each channel

To calibrate the card, run the setup program and follow the screen prompts. No attempt at calibration should be made in noisy locations or with a noisy calibration setup.

Each DAC output is available between the Analog Ground Pins (located on each end along the top edge of the card) and the Channel Test Point Pins located between each DAC channel's set of calibration potentiometers.

Note: After changing a channel's voltage range or polarity, the channel may require recalibration for best accuracy.

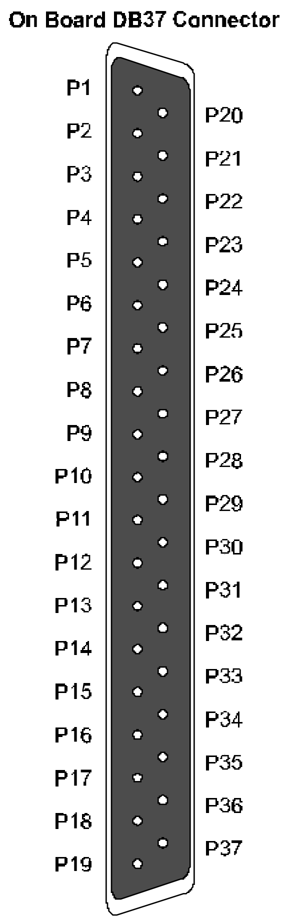
This page intentionally left blank

Chapter 8: Connector Pin Assignments

The analog outputs are accessible via a female 37-pin D type connector that extends through the back of the computer case and a DB37P solder cup plug may be used to make connections. Pin assignments are as follows:

Pin	Name	Function
1	D/A 0 Out	Analog DAC 0 Output
2	D/A 1 Out	Analog DAC 1 Output
3	D/A 2 Out	Analog DAC 2 Output
4	D/A 3 Out	Analog DAC 3 Output
5	D/A 4 Out	Analog DAC 4 Output
6	D/A 5 Out	Analog DAC 5 Output
7	D/A 6 Out	Analog DAC 6 Output
8	D/A 7 Out	Analog DAC 7 Output
9	D/A 8 Out	Analog DAC 8 Output
10	D/A 9 Out	Analog DAC 9 Output
11	D/A 10 Out	Analog DAC 10 Output
12	D/A 11 Out	Analog DAC 11 Output
13	D/A 12 Out	Analog DAC 12 Output
14	D/A 13 Out	Analog DAC 13 Output
15	D/A 14 Out	Analog DAC 14 Output
16	D/A 15 Out	Analog DAC 15 Output
17	+ 12 Vout	+12 VDC from PC
18	Analog GND	Analog Ground
19	-12 Vout	-12 VDC from PC
20	Return GND	Return Analog Ground
21	Return GND	Return Analog Ground
22	Return GND	Return Analog Ground
23	Return GND	Return Analog Ground
24	Return GND	Return Analog Ground
26	Return GND	Return Analog Ground
27	Return GND	Return Analog Ground
28	Return GND	Return Analog Ground
29	Return GND	Return Analog Ground
30	Return GND	Return Analog Ground
31	Return GND	Return Analog Ground
32	Return GND	Return Analog Ground
33	Return GND	Return Analog Ground
34	Return GND	Return Analog Ground
35	Return GND	Return Analog Ground
36	+5 Vout	+5 VDC from PC
37	Power GND	Power Ground

Note: The figure below shows how pins are numbered on D type connectors.



Chapter 9: Specifications

Analog Outputs

Resolution:	16 Binary bits (0 to 65535 decimal).
Channels:	16 or 8 Voltage output channels Voltage output ranges at 5mA max. 0.0 to 5.0 VDC. (76uV/bit) 0.0 to 10.0 VDC. -5.0 to +5.0 VDC. -10.0 to +10.0 VDC
Digital-to-Analog Converter:	AD660 D/A Converter, Double buffered / Simultaneous update
Relative Accuracy:	± 0.003%
Monotonicity:	15 bits over operating temperature range.
Settling Time:	8 usec to 0.0008% for full-scale step input.
Linearity:	± 1 LSB integral non-linearity over rated temperature range
Gain Stability:	15 ppm/°C
Output Drive Capability:	5mA minimum
Short-Circuit Current:	25 mA typical
Output Resistance:	Less than 0.1 W
Data Format:	16-bit binary

Power Requirements

+5 VDC at 2.5 A typical (16 channels installed)

±15 VDC are developed by internal DC-DC converter.

Environmental

Operating Temperature Range:	0 to +60 °C.
Storage Temperature Range:	-20 to +85 °C.
Humidity:	5% to 95% non-condensing.
Size:	Full size card, 13.310" long by 4.84" high

This page intentionally left blank

Appendix A: Sample Programs

Sample programs are provided on diskette with the AOBx/16. Sample Program #1 demonstrates general use of the card. This program prompts you for a voltage, calculates the closest actual voltage based on the 16-bit resolution of the DAC, and then programs the card to output this voltage. Sample Program #1 is provided in QuickBASIC, C, and Pascal.

Sample Program #2 will generate a sine, triangle, or sawtooth output waveform. This program is provided in QuickBASIC, C, and Pascal. A sample commented listing of the C language version is as follows (but refer to disk copies for the latest examples):

SAMPLE 2.C

This sample program will generate three different waveforms; sine, triangle, and sawtooth. You have the choice of base address, DAC number, and the number of points per cycle.

The base address entered during program execution should correspond to that set up on the card.

```
#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#define  PI 3.1415927
unsigned    counts;            /* number of points per cycle */
unsigned    baseadrs;          /* card base address */
unsigned    dacnum;            /* DAC used for output */
unsigned    progstruct[20000]; /* buffer to hold points */
```

Function: setparams() - local routine

Purpose: Prompts the user for DAC number, base address, and number of points per cycle.

Input : None

Calls: None

Output: None

```
void setparms()
{
clrscr();
printf(“Enter the base address of your card (in hex)\n”);
printf(“(Example: 300 : “);
scanf(“%x”,&baseadrs);
printf(“Enter the DAC number you wish to output to (0 or 1):”);
scanf(“%u”,&dacnum);
dacnum% = 2;
printf(“Enter the number of points that you wish to calculate per cycle,\n”);
```

```
printf("(20000 maximum, program will use modulus if needed);");
scanf("%u",&counts);
counts%=20001;
}  /end setparms*/
```

Function: sendtoport( ) - local routine

Purpose: Writes point buffer to the DAC until a key is pressed

Input: None

Calls: None

Output: None

```
void sendtoport()
{
int             i,temp;
long            j;
unsigned char   lowbyte,hibyte;

/*Each point is broken into the high byte and low byte, and then written to the
DAC in two separate bytes. */
do
   {
   for(i = 0; i <counts,i++)
      {
    temp = progstruct{i} % 256;
      lowbyte = (unsigned char)temp;
      temp = progstruct{i} / 256;
 hibyte = (unsigned char)temp;
    outportb(baseadrs+(dacnum*2),lowbyte);
    outportb(baseadrs+(dacnum*2+1),hibyte);
    }
    }
while (!kbhit());
outportb(baseadrs+(dacnum*2),0);     /*set DAC to 0 output */
outportb(baseadrs+(dacnum*2+1),0);
}                                     /*end sendtoport */
```

Function: sinecurve() - local routine

Purpose: Calculate the points to create a sine wave

Input: None

Calls: None

Output: None

```
void sinecurve()
{
int       i;
double    rads,sine;

if (counts == 0) return;                    /*no point -- no curve */

clrscr();
printf("Calculating sine wave points....");

rads = (double) 2 * PI / (counts - 1);             /* rad per count */

for(i = 0;i <counts;i++)
   {
   sine = (sin(rads * i) + 1.0) * 32767;
   progstruct[i] = (unsigned) sine;
   }
clrscr();
printf("Generating sine wave, press any key to stop....");
sendtoport();
}   /* end sinecurve */
```

Function: triaglecurve( ) - local routine

Purpose: Calculate the points to create a trinagle wave

Input:	None
Calls:	None
Output:	None

```
void trianglecurve(void)
{
int       i;
double    slope,temp;

if (counts == 0) return;       /* no counts -- no curve */

clrscr();
Printf("Calculating triangle wave points....");

slope = 65535.0 / counts * 2.0;       /* waveform slope */
for(i=0;i <counts/2;i++)
   {
   temp = slope * i;
   progstruct[i] = (int)temp;
```

```
   temp = 65535 - temp;
   progstruct[i+counts/2+1] = (int)temp;
   }
clrscr();
printf(“Generating triangle wave, press any key to stop....”);
sendtoport();
}                                           /* end triangle curve */
```

Function: sawcurve() - local routine

Purpose: Calculate the points to create a sawtooth wave

Input: None

Calls: None

Output: None

```
void sawcurve()
{
int       i;
double    slope,temp;

if (counts == 0) return;

clrscr();
printf(“Calculating sawtooth wave points....”);

slope = 4095.0 / counts;              /* sawtooth slope*/

for(i = 0,i <counts;i++)
   {
   temp = slope * i;
   progstruct[i] = (int) temp;
   }
clrscr();
printf(“Generating sawtooth wave, press any key to stop....”);
sendtoport();
}                                    /* end sawcurve */
```

Function: menulist() - local routine

Purpose: Display the menu choice on the screen

Input: None

Calls: None

Output: None

```
void menulist(void)
{
clrscr();
printf(“\n\n\n”);
printf(“Your menu selections are:\n”);
printf(“1. Input Card Data (do this first.)\n”):
printf(“2. Sine Curve\n”);
printf(“3. Triangle Curve\n”);
printf(“4. Sawtooth Curve\n”);
printf(“5. End Program, Return to DOS\n”);
printf(“Input Choice;”);
}                                            /* end menulist */
```

Function: main() - local routine

Purpose: Controls program execution

Input:	None
Calls:	None
Output:	None

```
void main(void)
{
char      menuchoice;
clrscr();
do
   {
   memset(progstruct, 0, sizeof(int); /* clear buffer  */
   menulist( );  /* display the menu*/
   menuchoice=getch( ); /* fetch the menu choice */
   switch(menuchoice)   /* execute menu selection*/
      {
 case ‘1’: setparms( ); /* fetch system parameters*/
    break;
 case ‘2’: sinecurve( );      /* generate a sine wave  */
 break;
 case ‘3’: trianglecurve( );   /* generate a triangle wave*/
 break;
 case ‘4’: sawcurve( ); /* generate a sawtooth wave*/
 break;
 case ‘5’: return;      /* exit to operating system */
 };
   }
while(1== 1);
}                                                   /* end main */
```

This page intentionally left blank

Declaration of Conformity

INDUSTRIAL COMPUTER SOURCE®

6260 Sequence Drive
San Diego, CA 92121-4371
(800) 523-2320

Industrial Computer Source declares under its own and full responsibility that the following products are compliant with the protection requirements of the 89/336/EEC and 73/23/EEC directives.

Only specific models listed on this declaration and labeled with the CE logo are CE compliant.

AOB8/16 AOB16/16

Conformity is accomplished by meeting the requirements of the following European harmonized standards:

EN 50081-1:1992 Emissions, Generic Requirements.
- EN 55022 Measurement of radio interference characteristics of information technology equipment.

EN 50082-1:1992 Immunity, Generic Requirements.
- IEC 801-2:1984 Immunity for AC lines, transients, common, and differential mode.
- IEC 801-3:1984 Immunity for radiated electromagnetic fields.
- IEC 801-4:1988 Immunity for AC and I/O lines, fast transient common mode.

EN 60950:1992 Safety of Information Technology Equipment.

Information supporting this declaration is contained in the applicable Technical Construction file available from:

INDUSTRIAL COMPUTER SOURCE EUROPE®
Z.A. de Courtaboeuf
16, Avenue du Québec
B.P. 712
91961 LES ULIS Cedex

August 29, 1997
San Diego, CA

Mr. Steven R. Peltier
President & Chief Executive Officer

BUG REPORT

While we have tried to assure this manual is error free, it is a fact of life that works of man have errors. We request you to detail any errors you find on this BUG REPORT and return it to us. We will correct the errors/problems and send you a new manual as soon as available. Please return to:

INDUSTRIAL COMPUTER SOURCE®

Attn: Documentation Department
P. O. Box 910557
San Diego, CA 92121-0557

Your Name: ____

Company Name: ____

Address 1: ____

Address 2: ____

Mail Stop: ______________________________

City: ______________________________________ State: _______ Zip: ____________

Phone: (_____) _______________

Product: **AOB8/16 & AOB16/16**

Manual Revision: **00650-010-6B**

Please list the page numbers and errors found. Thank you!