MODEL PCI-DIO-48(S)

USER MANUAL

FILE: MPCI-DIO-48S.D3c

Notice

The information in this document is provided for reference only. Portwell 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 Portwell, nor the rights of others.

IBM PC, PC/XT, and PC/AT are registered trademarks of the International Business Machines Corporation.

WARNING!!

ALWAYS CONNECT AND DISCONNECT YOUR FIELD CABLING WITH THE COMPUTER POWER OFF. ALWAYS TURN COMPUTER POWER OFF BEFORE INSTALLING A CARD. CONNECTING AND DISCONNECTING CABLES, OR INSTALLING CARDS INTO A SYSTEM WITH THE COMPUTER OR FIELD POWER ON MAY CAUSE DAMAGE TO THE I/O CARD AND WILL VOID ALL WARRANTIES, IMPLIED OR EXPRESSED.

Warranty

Prior to shipment, Portwell equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, Portwell assures its customers that prompt service and support will be available. All equipment originally manufactured by Portwell which is found to be defective will be repaired or replaced subject to the following considerations.

Terms and Conditions

If a unit is suspected of failure, contact Portwell' Customer Service department. Be prepared to give the unit model number, serial number, and a description of the failure symptom(s). We may suggest some simple tests to confirm the failure. We will assign a Return Material Authorization (RMA) number which must appear on the outer label of the return package. All units/components should be properly packed for handling and returned with freight prepaid to the Portwell designated Service Center, and will be returned to the customer's/user's site freight prepaid and invoiced.

Coverage

First Three Years: Returned unit/part will be repaired and/or replaced at Portwell option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, Portwell stands ready to provide on-site or in-plant service at reasonable rates similar to those of other manufacturers in the industry.

Equipment Not Manufactured by Portwell

Equipment provided but not manufactured by Portwell is warranted and will be repaired according to the terms and conditions of the respective equipment manufacturer's warranty.

General

Under this Warranty, liability of Portwell is limited to replacing, repairing or issuing credit (at Portwell discretion) for any products which are proved to be defective during the warranty period. In no case is Portwell liable for consequential or special damage arriving from use or misuse of our product. The customer is responsible for all charges caused by modifications or additions to Portwell equipment not approved in writing by Portwell or, if in Portwell opinion the equipment has been subjected to abnormal use. "Abnormal use" for purposes of this warranty is defined as any use to which the equipment is exposed other than that use specified or intended as evidenced by purchase or sales representation. Other than the above, no other warranty, expressed or implied, shall apply to any and all such equipment furnished or sold by Portwell.

Table of Contents

Chapter 1: Introduction

Features

- 48 Bits of Digital Input/Output.
- Interrupt Generation on Input Change of State. (Model "48S")
- Change-of-state Interrupt Software Enabled in Six 8-Input Ports.(Model "48S")
- All 48 I/O Lines Buffered on the Board.
- I/O Buffers Can Be Enabled/Disabled under Program Control.
- Four and Eight Bit Ports Independently Selectable for I/O.
- Pull-Ups on I/O Lines.
- +5V Supply Available to the User.
- Compatible with Industry Standard I/O Racks like Gordos, Opto-22, Potter & Brumfield, Western Reserve Controls, etc.

Applications

- Automatic Test Systems.
- Laboratory Automation.
- Robotics.
- Machine Control.
- Security Systems, Energy Management.
- Relay Monitoring and Control.
- Parallel Data Transfer to PC.
- Sensing Switch Closures or TTL, DTL, CMOS Logic.
- Driving Indicator Lights or Recorders.

These cards support 48 bits of parallel digital input/output on the PCI bus. They can be programmed to accept inputs or to provide outputs on two groups of three 8-bit ports. Further, in each group, one of the ports can be further divided into two four-bit nibbles.

The feature that distinguishes the "48S" model from the "48" card is that the state of all inputs can be monitored and, if any one or more bits change state, a latched interrupt request can be generated. Thus, it is not necessary to use software to continuously poll the inputs to detect a change of state. The change-of-state interrupt is disabled/enabled by a software write to an interrupt-enable register. Six bits in that register each control an eight-input port at one of two type 8255-5 Programmable Peripheral Interface chips. The change-of-state interrupt latch can be cleared by a software write.

Also, bit C3 at each 24-bit port can be used as an external interrupt to the computer if the IEN jumpers are installed. When bit C3 goes high (edge triggering), an interrupt is requested. Interrupts from the ports are OR'ed together and OR'ed with the change-of-state interrupt. Interrupt levels are assigned by the system.

The card was designed for industrial applications and can be installed in 7", or longer, PCI slots of IBM or compatible computers. Each I/O line is buffered and capable of sourcing 15 mA or sinking 24mA (64 mA on request). The card contains two Programmable Peripheral Interface chips type 8255-5 (PPI) to provide computer interface to 48 I/O lines. Each PPI provides three 8-bit ports A, B, and C. Each 8-bit port can be software configured to function as either inputs or output latches. Port C can also be configured as four

inputs and four output latches. Pull-ups on the card assure that there are no erroneous outputs at power up until the card is initialized by system software.

Tristate I/O line buffers (74LS245) are configured automatically by hardware logic for input or output use according to direction assignment from a control register in the PPI. Further, if a jumper is properly placed on the card, the tristate buffers may be enabled/disabled under program control. (See the Option Selection section to follow.)

I/O wiring connections are via 50-pin headers on the board. Two flat I/O cables connect the card to termination panels. Also, this provides compatibility with OPTO-22, Gordos, Potter & Brumfield, et al module mounting racks. Every second conductor of the flat cables is grounded to minimize crosstalk between signals in the cables. If needed for external circuits, +5 VDC power is available on each I/O connector pin 49. If you use this power, we recommend that you include a 1A fast-blow fuse in your circuits in order to avoid possible damage to the host computer or cable in the event of a malfunction in those external circuits.

The card occupies sixteen bytes of I/O address space. The base address is selected by the system. An illustrated setup program is provided on diskette with the card. Interactive displays show locations and proper settings jumpers to set up the interrupt enable function. Also, sample programs in Turbo-C and Turbo-Pascal are presented in the Software section of this manual.

Specification

Digital Inputs (TTL Compatible)

- Logic High: 2.0 to 5.0 VDC.
- Logic Low: -0.5 to +0.8 VDC.
- Input Load (Hi): 20 uA.
- Input Load (Lo): -200 uA.

Digital Outputs

- Logic High: 2.0 VDC min., source 32 mA.
- Logic Low: 0.55 VDC max., sink 64 mA.

- Power Output: +5 VDC from computer bus (ext. 1A fast-blow fuse recommended).
- Power Required: +5 VDC at 250 mA typical.
- Size: 6.9" Long.

Environmental

- Operating Temperature: 0 °C. to 60 °C.
- Storage Temperature: -50 °C. to +120 °C.
- Humidity: 0 to 90% RH, non-condensing.

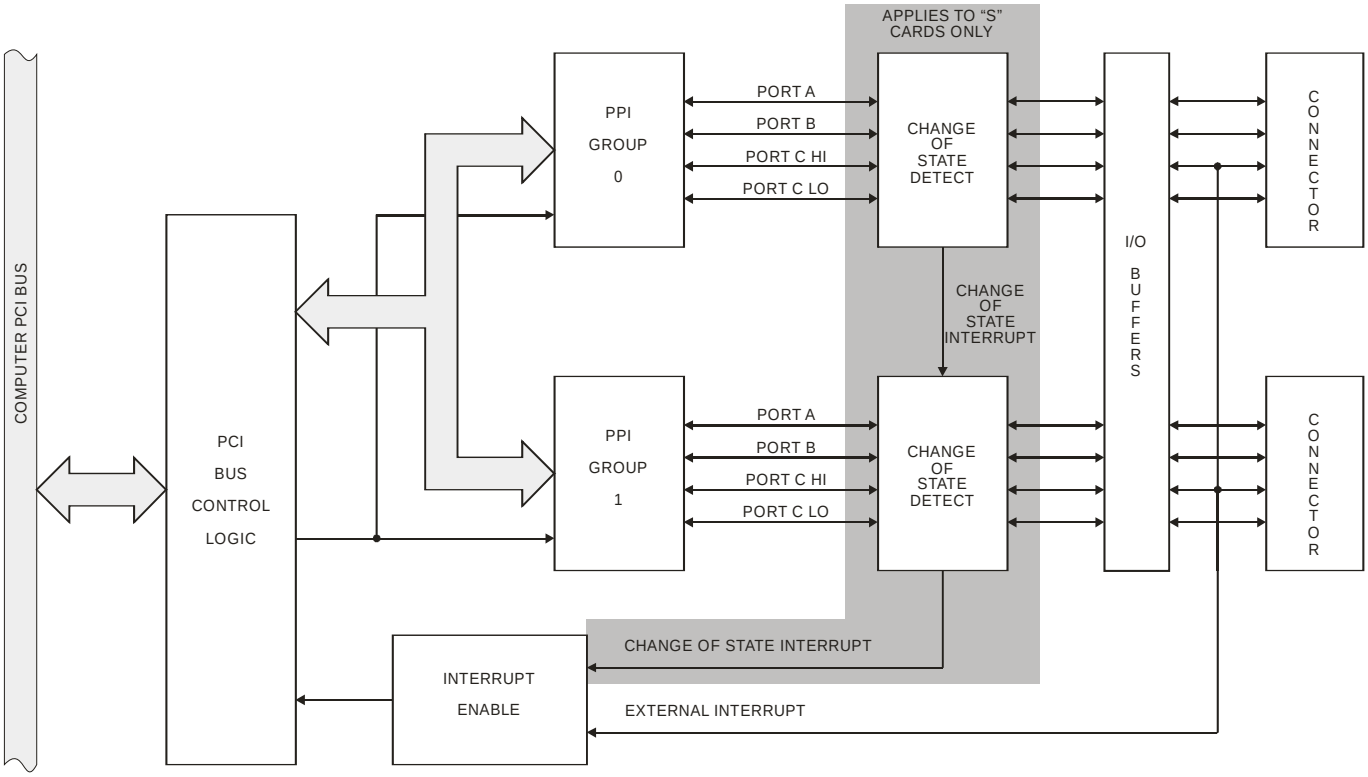

Figure 1-1: Block Diagram

Chapter 2: Installation

A printed Quick-Start Guide (QSG) is packed with the card for your convenience. If you've already performed the steps from the QSG, you may find this chapter to be redundant and may skip forward to begin developing your application.

The software provided with this card is on CD and must be installed onto your hard disk prior to use. To do this, perform the following steps as appropriate for your operating system.

Configure Card Options via Jumper Selection
Before installing the card into your computer, carefully read Chapter 3: Option Selection of this manual, then configure the card according to your requirements. Our Windows based setup program can be used in conjunction with Chapter 3 to assist in configuring jumpers on the card, as well as provide additional descriptions for usage of the various card options.

CD Software Installation

The following instructions assume the CD-ROM drive is drive "D". Please substitute the appropriate drive letter for your system as necessary.

DOS
1. Place the CD into your CD-ROM drive.
2. Type D : Enter to change the active drive to the CD-ROM drive.
3. Type I N S T A L L Enter to run the install program.
4. Follow the on-screen prompts to install the software for this board.

WINDOWS
1. Place the CD into your CD-ROM drive.
2. The system should automatically run the install program. If the install program does not run promptly, click START | RUN and type D : I N S T A L L, click OK or press Enter.
3. Follow the on-screen prompts to install the software for this board.

LINUX
1. Please refer to linux.htm on the CD-ROM for information on installing under linux.

Caution! * ESD A single static discharge can damage your card and cause premature failure! Please follow all reasonable precautions to prevent a static discharge such as grounding yourself by touching any grounded surface prior to touching the card.

Hardware Installation

1. Make sure to set switches and jumpers from either the Option Selection section of this manual or from the suggestions of SETUP.EXE.
2. Do not install card into the computer until the software has been fully installed.
3. Turn OFF computer power AND unplug AC power from the system.
4. Remove the computer cover.
5. Carefully install the card in an available 5V or 3.3V PCI expansion slot (you may need to remove a backplate first).
6. Inspect for proper fit of the card and tighten screws. Make sure that the card mounting bracket is properly screwed into place and that there is a positive chassis ground.
7. Install an I/O cable onto the card's bracket mounted connector.
8. Replace the computer cover and turn ON the computer which should auto-detect the card (depending on the operating system) and automatically finish installing the drivers.
9. Run PCIfind.exe to complete installing the card into the registry (for Windows only) and to determine the assigned resources.
10. Run one of the provided sample programs that was copied to the newly created card directory (from the CD) to test and validate your installation.

The base address assigned by BIOS or the operating system can change each time new hardware is installed into or removed from the computer. Please recheck PCIFind or Device Manager if the hardware configuration is changed. Software you write can automatically determine the base address of the card using a variety of methods depending on the operating system. In DOS, the PCI\SOURCE directory shows the BIOS calls used to determine the address and IRQ assigned to installed PCI devices. In Windows, the Windows sample programs demonstrate querying the registry entries (created by PCIFind and NTIOPCI.SYS during boot-up) to determine this same information.

INPUT/OUTPUT CONNECTIONS

To ensure that there is minimum susceptibility to EMI and minimum radiation, it is important that the card mounting bracket be properly screwed into place and that there be a positive chassis ground. Also, proper EMI cabling techniques (cable connect to chassis ground at the aperture, shielded twisted-pair wires, etc) should be used for the input/output wiring.

Chapter 3: Option Selection

Refer to the setup programs on the CD provided with the card. Also, refer to Figure 1-1, Block Diagram and Figure 3-1, Option Selection Map when reading this section of the manual.

External Interrupts are accepted on the I/O connector pin 9 (bit C3) for each group. The Interrupt signal is positive true. External Interrupts are enabled if the IEN0 (for Group 0) and IEN1 (for Group 1) jumper is installed. Interrupts are directed to an available IRQ level by the system.

A means of enabling or disabling the 74LS245 input/output buffers under program control is provided at the jumper position labeled TST/BEN. When the jumper is in the BEN (Buffer Enable) position, the I/O buffers are always enabled. When the jumper is in the TST (Tristate) position, enabled/disabled state is controlled by a control register. (See the programming section of this manual for a description.)

An LED, CR1, at the top left of the card to assist you in program development. Each time an interrupt is generated, the LED will illuminate and remain ON until the interrupt is reset. If there is an immediate reset of the interrupt, it is likely that the LED will not remain ON long enough to be observed.

Note

A jumper must be installed in either the TST or the BEN position for the card to function.

The foregoing are the only manual setups necessary to use either Input/Output selection. The change-of-state Interrupt Disable/Enable is done via software by writing to a control register in each PPI as described in Chapter 6, Programming.

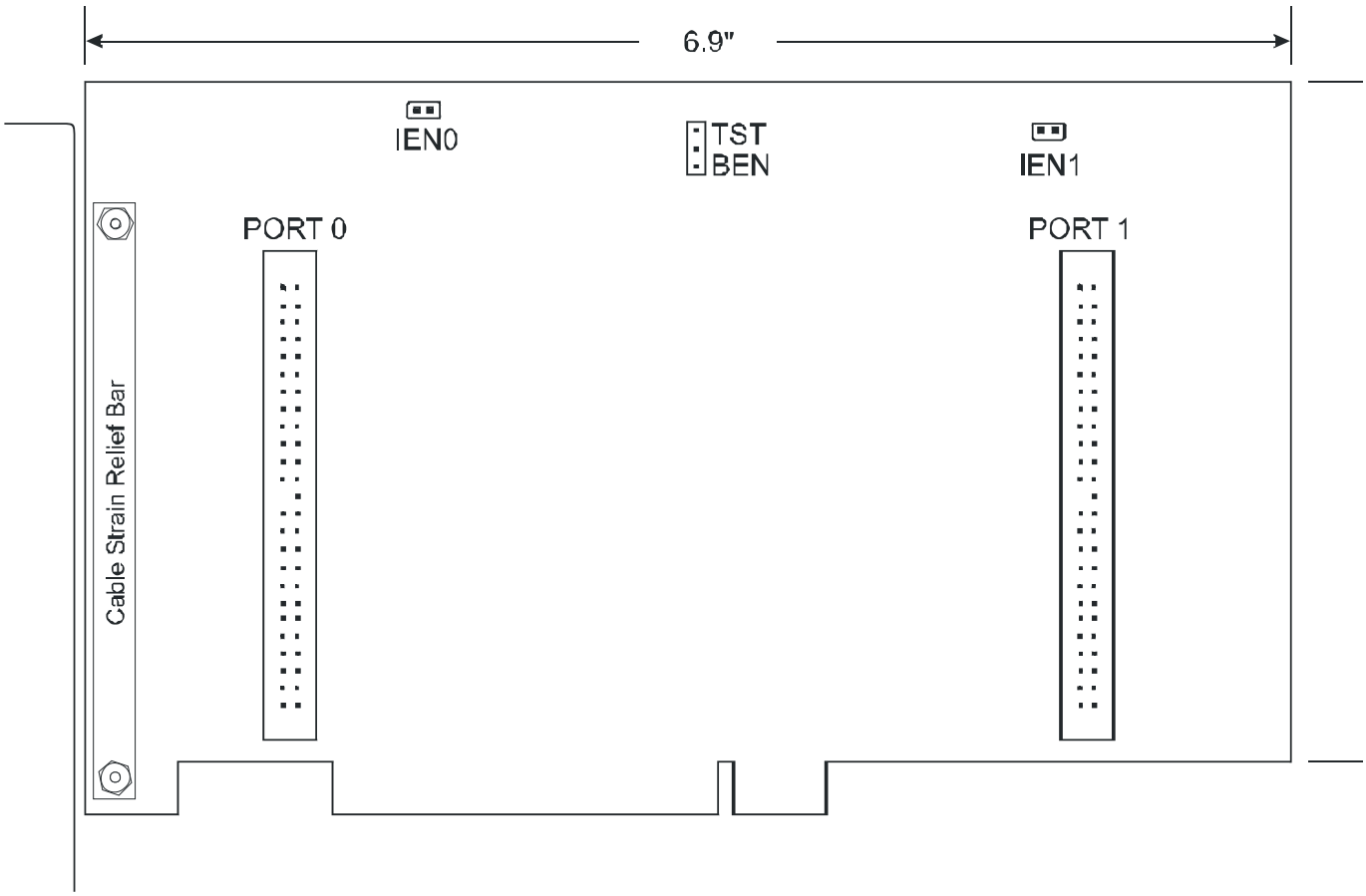

Figure 3-1: Option Selection Map

Chapter 4: Address Selection

These cards use one address space, occupying sixteen consecutive register locations.

PCI architecture is inherently plug-and-play in nature. This means that the BIOS or Operating System determines the resources assigned to PCI cards rather than you selecting those resources with switches or jumpers. As a result, you cannot set or change the card's base address or IRQ level. You can only determine what the system has assigned.

To determine the base address that has been assigned, run the PCIFind.EXE utility program. This utility will display a list of all of the cards detected on the PCI bus, the addresses assigned to each function on each of the cards, and the respective IRQs allotted.

Alternatively, some operating systems (Windows95/98/2000) can be queried to determine which resources were assigned. In these operating systems, you can use either PCIFind or the Device Manager utility from the System Properties Applet of the control panel. The cards are installed in the Data Acquisition class of the Device Manager list. Selecting the card, clicking Properties, and then selecting the Resources Tab will display a list of the resources allocated to the card.

PCIFind uses the Vendor ID and Device ID to search for your card, then reads the base address and IRQ assigned. If you want to determine these yourself, the Vendor ID is 494F (ASCII for "I/O") and the Device IDs are:

48: 0C60
48S: 0E60

The PCI bus supports 64K of I/O address space, so your card's addresses may be located anywhere in the 0000 to FFFF hex range.

Chapter 5: Software

Several programs are provided to support these Digital I/O cards and, also, to help you develop your application's software. Most of these programs are described on page 1-1 of this manual. The following paragraphs describe the IRQCOS driver, Win32COS driver (applicable for 48S only), setup program, and the VisualBASIC utility program

IRQCOS Driver

Description

IRQCOS.VXD is a Virtual Device Driver, or VxD, for Windows 95/98. IRQCOS.SYS is a Device Driver for Windows NT 4.0. Functionally, they are identical, and for the purposes of this manual the term IRQCOS Driver will be used to refer to the appropriate file for your operating system. Together with Win32COS.DLL, the IRQCOS Driver allows a program to easily respond to hardware change-of-state interrupt requests, or IRQs, and read data off the card following an IRQ.

Installation

To install IRQCOS.VXD in Windows 95/98, copy it to the directory C:\Windows\System. If your hard drive has a letter other than C, substitute the appropriate letter. It is not necessary to reboot your system prior to using IRQCOS.VXD.

To install IRQCOS.SYS in Windows NT, copy it to the directory C:\WINNT\System32\Drivers. If your hard drive has a letter other than C, substitute the appropriate letter. It is not necessary to reboot your system prior to using IRQCOS.SYS.

Win32COS.DLL

Description

Win32COS.DLL is a Dynamic Link Library, or DLL, for Windows 95/98 and NT. It provides a simple interface to the IRQCOS Driver, allowing a program to easily respond to hardware change-of-state interrupts and read data off the card after an IRQ occurs. Two functions are exported by Win32COS.DLL: InitCOSDriver, and GetCOSData. To use these functions in a program, they must first be imported into that program. The simplest way to import these functions in Pascal is to include the Win32COS unit in your Uses clause, or, in C, to include Win32COS.H. Additionally, in C, the library file CBCOS.LIB (for C++ Builder) or VCCOS.LIB (for Visual C++) must be linked to the program. This is accomplished by adding it to the project or editing the makefile. If you are not using the Win32COS header file and wish to import the functions manually, the necessary lines in Pascal would be:

```
function InitCOSDriver(BaseAddress: DWORD; IRQ: BYTE; BusType: SmallInt;
BusNumber:BYTE): ByteBool; cdecl; external 'Win32COS.dll';
function GetCOSData(Buf: Pointer): ByteBool; cdecl; external 'Win32COS.dll';
```

To import the functions manually in C, the necessary lines would be:

```
extern "C" __declspec(dllimport) unsigned char InitCOSDriver(unsigned long
BaseAddress, unsigned char IRQ, short BusType, unsigned char BusNumber);
extern "C" __declspec(dllimport) unsigned char GetCOSData(void *Buf);
```

InitCOSDriver

This function performs the initialization required to respond to change-of-state IRQs. It must be called before GetCOSData can be called, although calling GetCOSData first will only cause it to return a value of FALSE. The function's parameters are the card's Base Address, the IRQ level to monitor, the card's bus type, and the card's bus number. The last two parameters, BusType and BusNumber, are only used in Windows NT. The header files Win32COS.PAS (for Pascal) and Win32COS.H (for C) contain the enumerated constants that can be passed in the BusType parameter. The two possible values are PCIBus and Isa. For Windows 95/98, the IRQCOS Driver ignores these parameters and their values can be set to zero. The return result is TRUE if initialization was completed successfully, FALSE if not.

GetCOSData

This function suspends the thread that it was called from and waits for an IRQ to occur on the IRQ level that was passed to InitCOSDriver. GetCOSData will return immediately, however, if an IRQ occurred anytime before the GetCOSData call, as long as InitCOSDriver was previously called. Otherwise, GetCOSData will not return until an IRQ occurs. If the program needs to continue running while waiting for GetCOSData, a separate thread should be created from which to call GetCOSData. Immediately upon detecting an IRQ, GetCOSData reads 48 bits of data off the card and returns the data in a pointer supplied by the calling application. This pointer must be at least six bytes and must be allocated by the calling application. For cards that contain less than 48 bits of data, the lowest bits are valid and the extraneous data should be discarded. For example, a card that contains 24 bits of data will return the valid data in the lowest 3 bytes of the memory block pointed to by the supplied pointer, while the upper 3 bytes will be invalid. If multiple interrupts occur before any GetCOSData call, the data read after the most recent IRQ will be returned. The return result is TRUE if an IRQ was detected on the given IRQ level and data was successfully read, FALSE if InitCOSDriver was not called before calling GetCOSData.

SETUP.EXE

This program is supplied in the root or base directory as a tool for you to use in configuring jumpers on the card. It is menu-driven and provides pictures of the card on the computer monitor. You make simple keystrokes to select functions. The picture on the monitor then changes to show how the jumper should be placed to effect your choices.

The setup program is a stand-alone program that can be run at any time. It does not require that the card be plugged into the computer for any part of the setup. The program is self-explanatory with operation instructions and on-line help.

To run this program, at the DOS prompt, enter SETUP.EXE followed by Enter.

VisualBASIC Utility Driver

Extensions to the VisualBASIC 3.0 language are also included on the diskette provided with your card. (For later versions of VisualBASIC, use the ACCES32 driver and samples.) 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.

Chapter 6: Programming

These cards are I/O-mapped devices that are easily configured from any language and any language can easily perform digital I/O through the card's ports. This is especially true if the form of the data is byte or word wide. All references to the I/O ports would be in absolute port addressing. However, a table could be used to convert the byte or word data ports to a logical reference.

Developing Your Application Software

If you wish to gain a better understanding of the programs on diskette, then the information in the following paragraphs will be of interest to you. Refer to the data sheets and 8255-5 specification in Appendix A.

A total of 16 register locations are used by the 48(S). The PPIs are addressed consecutively with Address bits A3 through A0 as follows:

Address	Port Assignment	Operation
Base Address	PA Group 0	Read/Write
Base Address +1	PB Group 0	Read/Write
Base Address +2	PC Group 0	Read Write
Base Address +3	Control Group 0	Write Only
Base Address +4	PA Group 1	Read/Write
Base Address +5	PB Group 1	Read/Write
Base Address +6	PC Group 1	Read/Write
Base Address +7	Control Group 1	Write Only
Base Address +8	Enable/DisableBuffer,Grp0	Write Only
Base Address +9	Enable/DisableBuffer,Grp1	Write Only
Base Address +B	Enable Chg-of-St. Interrupt	Write Only
Base Address +F	Clear Chg-of-St. Interrupt	Write Only

Table 6-1: Address Assignment Table

These cards use two 8255-5 PPIs to provide a total of 48 bits input/output capability. The cards are designed to use each of these PPIs in Mode 0 wherein:

a. There are two 8-bit groups (A and B) and two 4-bit groups (C Hi and C Lo).
b. Any group can be configured as an input or an output.
c. Outputs are latched.
d. Inputs are not latched.

Each PPI contains a Control Register. This write-only, 8-bit register is used to set the mode and direction of the groups. At Power-Up or Reset, all I/O lines are set as inputs. Each PPI should be configured during initialization by writing to the Control Registers even if the groups are only going to be used as inputs. Output buffers are automatically set by hardware according to the Control Register states. Note that Control Registers are located at base address +3 and base address +7. Bit assignments in each of these Control Registers are as follows:

Bit	Assignment	Code
D0	Port C Lo (C0-C3)	1=Input, 0=Output
D1	Port B	1=Input, 0=Output
D2	Mode Select	1=Mode 1, 0=Mode 0
D3	Port C Hi (C4-C7)	1=Input, 0=Output
D4	Port A	1=Input, 0=Output
D5,D6	Mode Select	00=Mode 0, 01=Mode 1, 1x=Mode 2
D7	Mode Set Flag	1=Active

Table 6-2: Control Register Bit Assignment

Note

Mode 1 cannot be used by these cards without modification. Thus, bits D2, D5, and D6 should always be set to "0". If your card has been modified to operate in Mode 1, then there will be an Addendum page in the front of this manual. These cards cannot be used in Mode 2 of the PPI.

Note

In Mode 0, do not use the control register byte for the individual bit control feature. The hardware uses the I/O bits to control buffer direction on this card. The control register should only be used for setting up input and output of the ports and enabling the buffer.

These cards provide a means to enable/disable the tristate I/O buffers under program control. If the TST/BEN jumper on the card is installed in the BEN position, the I/O buffers are permanently enabled. However, if that jumper is in the TST position, enable/disable of the buffers is software controlled via the control register as follows:

a. The card is initialized in the receive mode by the computer reset command.

b. When bit D7 of the Control Register is set high, direction of the three groups of the associated PPI chip as well as the mode can be set. For example, a write to Base Address +3 with data bit D7 high programs port direction at Group 0 ports A, B, and C. If, for example, hex 80 is sent to Base Address +3, the Port 0 PPI will be configured in mode 0 with Groups A, B, and C as outputs.

At the same time, data bit D7 is also latched in a buffer controller for the associated PPI chip. A high state disables the buffers and, thus, all four buffers will be put in the tristate mode; i.e. disabled.

c. Now, if any of the groups are to be set as outputs, you may set the values to the respective group with the outputs still in the tristate condition. (If all groups are to be set as inputs, this step is not necessary.)

d. If data bit D7 is low when the control byte is written, ONLY the associated buffer controller is addressed. If, for example, a control byte of hex 80 has been sent as previously described, and the data to be output are correct, and it is now desired to open the three groups, then it is necessary to send a control byte of hex 00 to base address +3 to enable the port 0 buffers. When you do this, the buffers will be enabled.

Note

Note that all data bits except D7 must be the same for the two control bytes

Those buffers will now remain enabled until another control byte with data bit D7 high is sent to base address +3.

Similarly, the Port 1 groups can be enabled/disabled via the control register at base address +7. The following program fragment in C language illustrates the foregoing:

```
const BASE_ADDRESS 0x300;
        outportb(BASE_ADDRESS +3, 0x89);    /*This instruction sets the mode to Mode 0, ports
                                            A and B as output, and port C as input. Since bit D7 is
                                            high, the output buffers are set to tristate condition. See
                                            item b. above.*/
outportb(BASE_ADDRESS,0);
        outportb(BASE_ADDRESS+1,0);         /*These instructions set the initial state of ports A
                                            and B to all zeroes. Port C is not set because it is
                                            configured as an input. See item c. above.*/
        outportb(BASE_ADDRESS +3, 0x09);    /*Enable the tristate output buffers by using the
                                            same control byte used to configure the PPI, but now set
                                            bit D7 low. See item d. above.*/
```

Programming Example (BASIC)

The following example in BASIC is provided as a guide to assist you in developing your working software. In this example, the card base address is 2D0 hex and the I/O lines of group 0 are to be setup as follows:

Port A	=	Input
Port B	=	Output
Port C Hi	=	Input
Port C Lo	=	Output

The first step is to configure the control register. Configure bits of the control register as:

D7	1	Active Mode Set
D6	0	Mode 0
D5	0	Mode 0
D4	1	Port A = Input
D3	1	Port C Hi = Input
D2	0	Mode 0
D1	0	Port B = Output
D0	0	Port C Lo = Output

This corresponds to 98 hex. If the card address is 2D0 hex, use the BASIC OUT command to write to the control register as follows:

```
10      BASEADDR=&H2D0
20      OUT BASEADDR+3,&H98
```

To read the inputs at Port A and the upper nybble of Port C:

```
30      X=INP(BASEADDR)             'Read Port A
40      Y=INP(BASEADDR+2)/16        'Read Port C Hi
```

To set outputs high (1) at Port B and the lower nybble of Port C:

```
50      OUT BASEADDR+1,&HFF     'Turn on all Port B bits
60      OUT BASEADDR+2,&HF      'Turn on all bits of Port C
lower nybble
```

Enabling/Disabling I/O Buffers

When using the tristate mode (Jumper in the TST position), the method to disable the I/O buffers involved writing a control word to the Control Register at Base Address +3 and Base Address +7. This control word was required to have bit D7 (the most significant bit) set. That meant that the PPI translated it as an "active mode set" and reset the output data latches to "zero" on all output ports and the output buffers were disabled. However, if the buffers are to be enabled at a later time, the output latches will be in a "zero" state. For example, if all the outputs were 1's, they will now be 0's and the output buffers will be disabled. This problem can be resolved as follows.

Two computer I/O bus addresses are available that permit you to enable or disable the I/O buffers at will, without programming the PPI mode. Buffers for Port 0 bits are enabled/disabled at Base Address +8 and buffers for Port 1 bits are enabled/disabled at Base Address +9. To enable the buffers and to set outputs to the desired state, you can write to the Control Register with bit D7 low. If you wish to subsequently disable the buffers, you can write to the Control Register with bit D7 high. In this way you can enable/disable the output buffers without programming the PPI mode.

Note

When writing a command byte to these cards while the TST jumper is installed, the PPI output buffers are disabled. Thus, when you desire to to change the mode, you must first set the new mode and then enable the buffers. Enabling the buffers can be done at either Base Address +3 (or +7) or Base Address +8 (or +9).

Change-of-State Interrupts

At Power-up or Reset, a latch disables all IRQ sources on the card. In order to properly disable/enable interrupts, you must program the Change-Of-State Interrupt Enable Register first. To program this Change-of-State-Interrupt-Enable Register, write to it at Base Address+B. Data bits D0 through D5 control ports A, B, and C of the 8255 PPIs as shown in Table 4. Any access of Base+B will enable the non-COS IRQ associated with
port C bit 3.

Bit	Port Controlled
D0	Group 0, Port A
D1	Group 0, Port B
D2	Group 0, Port C
D3	Group 1, Port A
D4	Group 1, Port B
D5	Group 1, Port C

Table 6-3: Change-of-state-interrupt-enable Register

Writing a "one" disables the port; writing a "zero" enables it. When IRQs occur the interrupt state is latched. To clear the latch, write anything to Base Address+F.

Chapter 7: Connector Pin Assignments

Two 50-pin headers are provided on these cards: one for each 24-bit I/O group. The mating connector is an AMP type 1-746285-0 or equivalent. Connector pin assignments are listed below. Notice that every second line is grounded to minimize crosstalk between signals.

Assignment		Pin		Assignment	Pin
Port C Hi	PC7	1		Ground	2
Port C Hi	PC6	3		"	4
Port C Hi	PC5	5		"	6
Port C Hi	PC4	7		"	8
Port C Lo	PC3*	9		Ground	10
Port C Lo	PC2	11		"	12
Port C Lo	PC1	13		"	14
Port C Lo	PC0	15		"	16
Port B	PB7	17		Ground	18
Port B	PB6	19		"	20
Port B	PB5	21		"	22
Port B	PB4	23		"	24
Port B	PB3	25		"	26
Port B	PB2	27		"	28
Port B	PB1	29		"	30
Port B	PB0	31		"	32
Port A	PA7	33		Ground	34
Port A	PA6	35		"	36
Port A	PA5	37		"	38
Port A	PA4	39		"	40
Port A	PA3	41		"	42
Port A	PA2	43		"	44
Port A	PA1	45		"	46
Port A	PA0	47		"	48
+5 VDC		49		Ground	50

* This line is an I/O port and also a User Interrupt.

Table 7-1: Connector Pin Assignments

Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: ***tech@portwell.com.*** Please detail any errors you find and include your mailing address so that we can send you any manual updates.