# Model PIOD24
# Product Manual

**MANUAL NUMBER : 00650-141-1A**



*6260 SEQUENCE DRIVE, SAN DIEGO, CA 92121 (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.

# Guarantee

A thirty day money-back guarantee is provided on all **standard** products sold. **Special order products** are covered by our Limited Warranty, *however they may not be returned for refund or credit. 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.

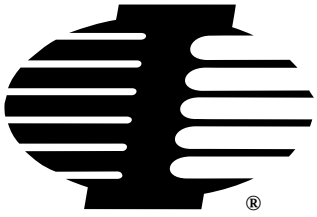
# Limited Warranty

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

**Current Revision 1A**

**September 1997**

# Chapter 1: Installation

## Backing up the Disk

The software provided with the PIOD24 PCMCIA card is supplied on a MS-DOS diskette. As with any software package, you should make back-up copies for everyday use and place your original master diskette in a safe location.

The easiest way to make a back-up copy is to use the DOS DISKCOPY utility.

In a single-drive system the command is diskcopy a: a:

In a two-drive system the command is diskcopy a: b:
(This will copy 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:

| | |
|---|---|
| PIDO24: | The root or base directory that contains the PIOD24.COM client driver for use with your laptop's existing PCMCIA socket driver. |
| 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 3.x |
| ACCES32: | Our utility driver kit that adds VBACCESS support for the 32-bit operating systems of Windows 95 and NT. |
| PSAMPLES: | Contains the PASCAL sample programs with source code and executable versions included. |
| CSAMPLES: | Contains the C language sample programs with source code and executable versions included. The sample programs are heavily commented and may be used to test the operation of the card. |

## Installing the Card

The PIOD24 card can be installed in any PCMCIA Type II card slot. There are no switches or jumpers to set. Everything on the PIOD24 card is programmable including address and interrupt level.

### DOS or Windows 3.X Installation

First install our software which will modify your AUTOEXEC.BAT file to add the call to load our client communications driver. You must also install the Card & Socket Services (CSS) software that was provided with your PCMCIA capable computer.

### DOS 'Plug N Play' OR Superclient Installation

DOS computers with a SuperClient installed may not require installing the PIOD24.COM program in the AUTOEXEC.BAT file. The SuperClient uses the PIOD24 internal CIS (configuration information) to configure the card and prepare system for use. Consult the SuperClient documentation to determine the installation procedure.

### Windows 95 Installation

Windows95 will automatically recognize the installation of the PCMCIA card and provide a list of options to install the card. Select the option "Driver from disk provided by hardware manufacturer" and select the diskette provided which will install support for the card. Do not run the install program unless to install the samples or programming support files. Depending on options selected in the Windows95 PCMCIA drivers, you may hear a "rising" two-note tone upon successful installation and a new PCMCIA icon in the taskbar tray.

Turn the power off and install the PIOD24 card into the PCMCIA slot of your computer.

**CAUTION:**

The PIOD24 must be oriented with the label side up, or "side A" in PCMCIA terms. The standard PCMCIA case is "keyed" to prevent improper insertion but may be broken off if excessive force is used during installation. **DO NOT FORCE THE PIOD24 CARD INTO THE SLOT.**

Install the card with the label facing "up" or towards side "A" which is usually towards the keyboard, but you should check with your systems owners manual. When correctly oriented, the PIOD24 card should seat in the slot with little pressure. Force the PIOD24 card in upside down and you will probably burn out components as well as destroy the key slot.

Reboot your computer after installing the PIOD24 so the drivers are loaded into memory. You may then run the SAMPLE1 software to confirm operation of your PIOD24 card.

# Chapter 2: Functional Description

**Features**

- Type II PCMCIA Card
- Plug-N-Play Windows95 Installation
- 24 Bit Digital I/O port
- Four and eight-bit groups independently selectable for digital I/O
- Three 16-bit 10MHz counters
- Software setup including FINDBASE and TSR setup program
- 37 pin D-sub Male Connector via cable adaptors
- Supports external counter clocks, gates and outputs, and internal control.

## Introducing PCMCIA

PCMCIA started as a way to add extra memory to laptop and portable computers. As the market grew, the absence of a standard expansion slot was recognized and the second release added hardware support for I/O devices. The PIOD24 is one of a family of PCMCIA devices offered that extend the ports and functionality of PC systems. This expansion port may be referred on your computer system as either a PCMCIA or PC CARD port.

## Software Compatibility

Industrial Computer Source supplies drivers and example programs to provide the PIOD24 with complete access to a the PIOD24 variety of programming languages and operating systems; including DOS, 16-bit and 32-bit Windows (Windows 3.x vs Windows 95 and Windows NT).

## Card and Socket Services

PCMCIA Card and Socket services must be loaded on the host computer system before using the PIOD24 card and is typically supplied by the PCMCIA slot provider (the laptop or adaptor manufacturer) to provide the software interface to the card slot drive. In addition, our TSR memory-resident program, described below, is required to configure the card once it is recognized by the card and socket services.

The Card and Socket services handle the PCMCIA slot software interface with your operating system when a new card is inserted (or removed) or when power is applied (or upon shut-down). Once the PCMCIA Card and Socket Services recognizes the PIOD24 card it will then appear to your application software like a card on the internal ISA bus.

## Calibration & Service

No calibration is required for PCMCIA devices and especially for the all-digital PIOD24. The case may not be opened and there are no parts inside which you can service. There are no socketed components. Opening the PIOD24 case will void your warranty.

## Utility Software

We include programs on 3.5" diskette to support the card and to help you develop your application program(s). The card supports I/O bus addresses from 000 to 3FF (hex) and a program called FINDBASE helps you find an open base address to use with the card.

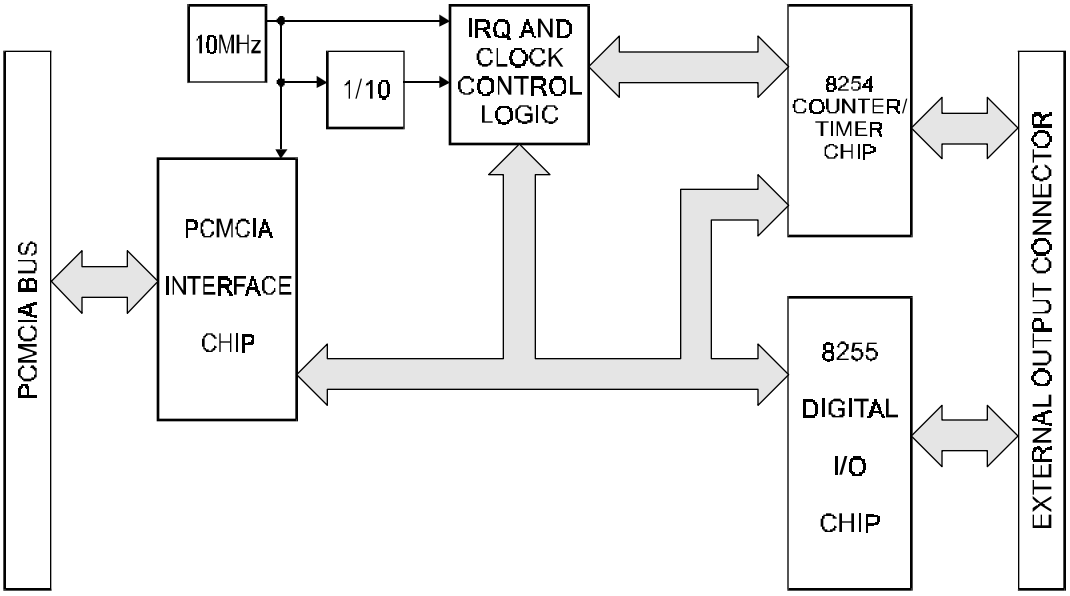
The TSR client program has two required command-line parameters (entered in hexadecimal format); the base I/O address and the interrupt request (IRQ) number. Optionally, a “verbose” switch will turn on a detailed report of the communication between the TSR and the PCMCIA card for diagnostic troubleshooting. A value of zero for the IRQ number will configure the card not to use any interrupts.

| | |
|---|---|
| Syntax: | PIOD24 [Base Address in Hex] [IRQ number in Hex] (verbose) |
| For example: | `PIOD24 300 0A verbose` |

Would configure the PIOD24 card at a base address of 300 (hex) and IRQ at 10 (decimal). If no values are used with the TSR program then the following error message is displayed:

```
The command line parameters were incorrect.
syntax: piod24 <port address> <IRQ>
example:piod24 120 5
example:piod24 300 0A
```

**PIOD24 Block Diagram**



This page intentionally left blank

# Chapter 3: Cable Connections

The PIOD24 requires our CAB-PIOD cable to provide the transition between the microminiature 32 pin PCMCIA connector and a standard 37-pin D subminiature connector to interface to your project. The 37-pin connector is equipped with 4-40 threaded standoffs (female screw lock) to provide strain relief. Pins 34-37 of the cable are not connected.

| **PIOD24 Connector** | | **CAB-PIOD Cable** |
|---|---|---|
| Pin | Description | Pin Standard DB-37 Connector |
| 1 | Digital Port A Bit 0 | 1 |
| 2. | Digital Port A Bit 1 | 2 |
| 3. | Digital Port A Bit 2 | 3 |
| 4. | Digital Port A Bit 3 | 4 |
| 5. | Digital Port A Bit 4 | 5 |
| 6. | Digital Port A Bit 5 | 6 |
| 7. | Digital Port A Bit 6 | 7 |
| 8. | Digital Port A Bit 7 | 8 |
| 9. | Digital Port B Bit 0 | 9 |
| 10. | Digital Port B Bit 1 | 10 |
| 11. | Digital Port B Bit 2 | 11 |
| 12. | Digital Port B Bit 3 | 12 |
| 13. | Digital Port B Bit 4 | 13 |
| 14. | Digital Port B Bit 5 | 14 |
| 15. | Digital Port B Bit 6 | 15 |
| 16. | Digital Port B Bit 7 | 16 |
| 17. | Digital Port C Bit 0 | 17 |
| 18. | Digital Port C Bit 1 | 18 |
| 19. | Digital Port C Bit 2 | 19 |
| 20. | Digital Port C Bit 3 | 20 |
| 21. | Digital Port C Bit 4 | 21 |
| 22. | Digital Port C Bit 5 | 22 |
| 23. | Digital Port C Bit 6 | 23 |
| 24. | Digital Port C Bit 7 | 24 |
| 25. | Ground (Tied to pin 33) | 25 |
| 26. | Counter 0 Output | 26 |
| 27. | Counter 0 Gate Input | 27 |
| 28. | Counter 0 External Clock Input | 28 |
| 29. | Counter 2 Output | 29 |
| 30. | Counter 2 Gate Input | 30 |
| 31. | Counter 2 External Clock Input | 31 |
| 32. | External Interrupt | 32 |
| 33. | Ground (Tied to pin 25) | 33 |
| 34. | No Connections | 34-37 |

This page intentionally left blank

# Chapter 4: Address Selection

The PIOD24 base address can be selected anywhere within an I/O address range 000-3FF hex, providing that the addresses do not overlap with other functions. If in doubt, refer to the table below for a list of standard address assignments or use the base address locator program FINDBASE provided on diskette will assist you to avoid an address conflict. (The primary and secondary binary synchronous communication ports are supported by the Operating System.)

| 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) |

**Standard Address Assignments for PC and PC/XT Computers**

This page intentionally left blank

# Chapter 5: Software

Several programs are supplied to support the PIOD24 Digital I/O Card and, also, to help you to develop your applications program. These programs are on a diskette that comes with your board.

## Sample Programs

- SAMPLE1 Demonstrates control of the 8254 and 8255 in two parts.

  The three counters may be individually set to any of its six modes and loaded with a 16-bit divisor. The user is prompted for the I/O address of the board, asked to press a key, and then is allowed to do it again.

  To manipulate the three digital ports the program asks the user to input the board's base I/O address and to select a port. The data direction is requested (input or output). If the port is set for input the byte value at the pins is displayed. If the port is set for output the user is prompted for a value to set the pins to and that value is displayed. The user is informed that he must press a key to continue.
- SAMPLE2 Checks the operation of the control register located at BASE+8 which selects alternate clock inputs and counter interrupt generation.
- VBACCES A 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 3.x.
- ACCES32 Provides the same functions as VBACCES for the Windows95 and WinNT 32-bit operating systems.

## VisualBasic Utility Driver

Extensions to the VisualBASIC language are provided on the diskette supplied 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 as 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 6: Programming

## Developing your Application Software

Once the Card & Socket Services (CSS) and Client Drivers are installed the PIOD24 card may be controlled by writing to and reading from the control registers.

### Port Addresses & Functions

All I/O accesses may be performed as bytes. The registers, in eight- bit format, are presented here:

**PIOD24 Register Address Selection**

| Address | Write Operation | Read Operation |
|---|---|---|
| Base Address | 82C55 Port A Set | Read back data |
| Base Address +1 | 82C55 Port B Set | Read back data |
| Base Address +2 | 82C55 Port C Set | Read back data |
| Base Address +3 | 82C55 Control Register | Write Only |
| Base Address +4 | 82C54 Counter 0 Load | Counter 0 Read back |
| Base Address +5 | 82C54 Counter 1 Load | Counter 1 Read back |
| Base Address +6 | 82C54 Counter 2 Load | Counter 2 Read back |
| Base Address +7 | 82C54 Counter Control | Write Only |
| Base Address +8 | Interrupt & Clock Source | Write Only |

## Digital I/O Port

The PIOD24 card supports operating mode 0 of the 82C55 PPI. Mode 0 is the most frequently used mode of operation wherein:

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

The control register at base address +3 is a write-only 8-bit register. It is used to set the mode and direction of the ports. At Power-Up or Reset, all I/O lines are set as inputs. The PPI should be configured during initializing by writing to the control register even if the ports are only going to be used as inputs.

Bit assignments in the control register are as follows:

**8255 Control Register Bit Assignments**

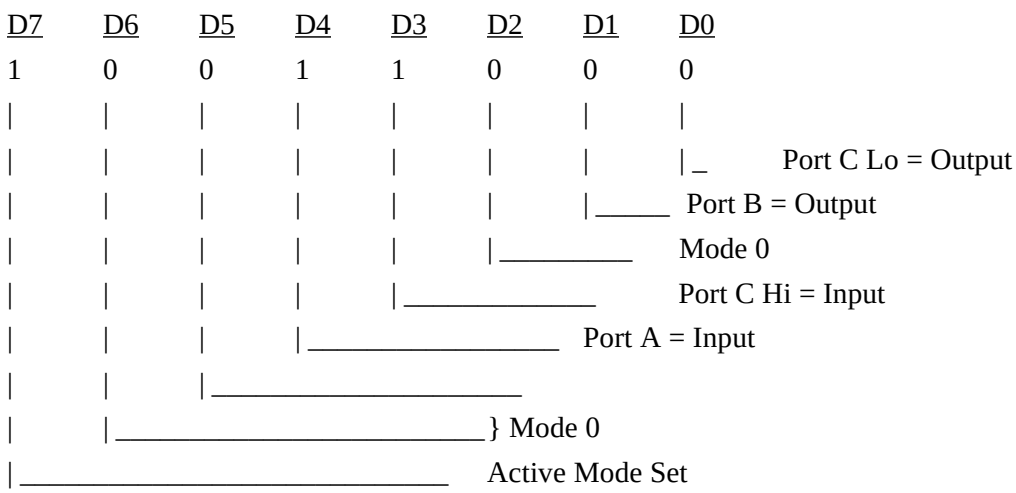
| Data Bit | Option Selected | | |
|---|---|---|---|
| D0 | Port C (C0-C3) | 1 = Input | 0 = Output |
| D1 | Port B | 1 = Input | 0 = Output |
| D2 | Mode Selection | 1 = Mode 1 | 0 = Mode 0 |
| D3 | Port C (C4-C7) | 1 = Input | 0 = Output |
| D4 | Port A | 1 = Input | 0 = Output |
| | | | |
| D5,D6 | Mode Selection | 00 = Mode 0 | |
| | | 01 = Mode 1 | |
| | | 1X = Mode 2 | |
| D7 | | Mode Set Flag | 1 = Active |

## Programming Example

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

| | | | |
|---|---|---|---|
| Port A | Input | Port C Hi | Input |
| Port B | Output | Port C Lo | Output |

Configure bits of the control register as follows:




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
```

## Programmable Interval Timer

The PIOD24 contains a type 82C54 programmable counter/timer which allows you to implement such functions as a Real-Time Clock, Event Counter, Digital One-Shot, Programmable Rate Generator, Square-Wave Generator, Binary Rate Multiplier, Complex Wave Generator, and/or a Motor Controller. The 82C54 is a flexible but powerful device that consists of three independent, 16-bit, presettable, down counters. Each counter can be programmed to any count as low as 1 or 2, and up to 65,535 in binary format, depending on the mode chosen.

## Operational Modes

The 82C54 modes of operation are described in the following paragraphs to familiarize you with the versatility and power of this device. For those interested in more detailed information, a full description of the 82C54 programmable interval timer can be found in the Intel (or equivalent manufacturers) data sheets. The following conventions apply for use in describing operation of the 82C54:

| | |
|---|---|
| Clock: | A positive pulse into the counter's clock input. |
| Trigger: | A rising edge input to the counter's gate input. |
| Counter Loading: | Programming of a binary count into the counter. |

### Mode 0: Pulse on Terminal Count

After the counter is loaded, the output is set low and will remain low until the counter decrements to zero. The output then goes high and remains high until a new count is loaded into the counter. A trigger enables the counter to start decrementing.

### Mode 1: Retriggerable One-shot

The output goes low on the clock pulse following a trigger to begin the one-shot pulse and goes high when the counter reaches zero. Additional triggers result in reloading the count and starting the cycle over. If a trigger occurs before the counter decrements to zero, a new count is loaded. Thus, this forms a re-triggerable one-shot. In mode 1, a low output pulse is provided with a period equal to the counter count-down time.

### Mode 2: Rate Generator

This mode provides a divide-by-N capability where N is the count loaded into the counter. When triggered, the counter output goes low for one clock period after N counts, reloads the initial count, and the cycle starts over. This mode is periodic, the same sequence is repeated indefinitely until the gate input is brought low. This mode also works well as an alternative to mode 0 for event counting.

### Mode 3: Square Wave Generator

This mode operates periodically like mode 2. The output is high for half of the count and low for the other half. If the count is even, then the output is a symmetrical square wave. If the count is odd, then the output is high for (N+1)/2 counts and low for (N-1)/2 counts. Periodic triggering or frequency synthesis are two possible applications for this mode. Note that in this mode, to achieve the square wave, the counter decrements by two for the total loaded count, then reloads and decrements by two for the second part of the wave form.

### Mode 4: Software Triggered Strobe

This mode sets the output high and, when the count is loaded, the counter begins to count down. When the counter reaches zero, the output will go low for one input period. The counter must be reloaded to repeat the cycle. A low gate input will inhibit the counter.

### Mode 5: Hardware Triggered Strobe

In this mode, 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 retriggerable. The output will not go low until the full count after the rising edge of the trigger.

## Counter/Timer Registers

**Base + 4 Write/Read: Counter#0** When writing, this register is used to load a counter value into the counter. The transfer is either a single or double byte transfer, depending on the control byte written to the counter control register at BASE ADDRESS + 7. If a double byte transfer is used, then the least-significant byte of the 16 bit value is written first, followed by the most significant byte. When reading, the current count of the counter is read. The type of transfer is also set by the control byte.

**Base + 5 Write/Read: Counter #1** See description for **Base + 4 Write/Read**.

**Base + 6 Write/Read: Counter#2** See description for **Base + 4 Write/Read**.

**Base + 7 Write: Counter Control Register** The control byte specifies the counter to be programmed, the counter mode, the type of read/write operation, and the modulus. The control byte format is as follows:

**Counter Control Byte**

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|
| SC1 | SC0 | RW1 | RW0 | M2 | M1 | M0 | BCD |

SC0-SC1: These bits select the counter modified by the counter control byte.

| SC1 | SC0 | Function |
|---|---|---|
| 0 | 0 | Program Counter 0 |
| 0 | 1 | Program Counter 1 |
| 1 | 0 | Program Counter 2 |
| 1 | 1 | Read Back Command |

RW0-RW1:These bits select the read/write mode of the selected counter.

| RW1 | RW0 | Counter Read/Write Function |
|---|---|---|
| 0 | 0 | Counter Latch Command |
| 0 | 1 | Read/Write LS Byte |
| 1 | 0 | Read/Write MS Byte |
| 1 | 1 | Read/Write LS Byte, then MS Byte |

M0-M2: These bits set the operational mode of the selected counter.

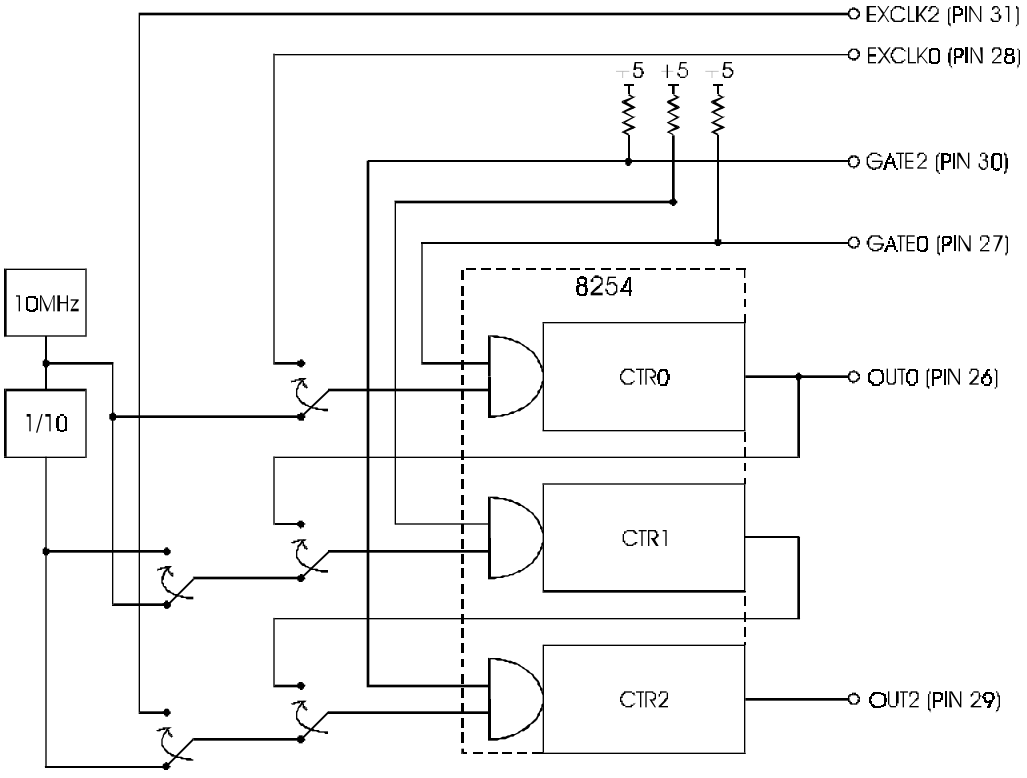
| MODE | M2 | M1 | M0 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | X | 1 | 0 |
| 3 | X | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |

BCD: Set the selected counter to count in Binary Coded Decimal or straight Binary. (BCD=1 selects BCD mode, BCD=0 selects straight binary coding).

**Base + 8 Write: Interrupt and Clock Source Control** Because of the limited number of pins on the 32 pin connector, all signals from all three counters are not available externally. This is compensated for by the ability to chain counters and control the count source. Bit 0 and 3 of port C are the outputs of the 82C55 programmed in Mode 1 or Mode 2. These two modes provide an interrupt control of closely coupled parallel interfaces. To program an 82C55 for Mode 1 or 2 you will need to consult the 82C55 data sheet.

## Counter Source Control

The counters on the PIDO24 have quite a bit of programmability. Due to the architecture of the board, there is additional counter source and chaining flexibility beyond the standard functions of the 82C54. The diagram shows schematically the programmable options.



The Interrupt and Clock Source Control byte format is as follows:

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|
| INT2 | INT1 | INT0 | CKSEL2 | CKSEL1 | CKSEL0 | CLK1 | CLK0 |

INT0-INT2: These bits select the Interrupt Source.

| **INT2** | **INT1** | **INT0** | **Interrupt Source** |
|---|---|---|---|
| 0 | 0 | 0 | No interrupts |
| 0 | 0 | 1 | External Interrupt |
| 0 | 1 | 0 | Interrupt from Bit 0 of Port C |
| 0 | 1 | 1 | Counter 0 |
| 1 | 0 | 0 | Counter 1 |
| 1 | 0 | 1 | Counter 2 |
| 1 | 1 | 0 | Interrupt from Bit 3 of Port C |

## Clock Source Select

These control register bits select the counter clock source for each counter. Selection of counter cascading other than as three 16-bit counters will override the counter source selection for the linked counters. The clock source bits may be set as follows:

| CKSEL0 | Counter 0 Clock Source |
|---|---|
| 0 | 10 MHz clock |
| 1 | External clock - applied at pin 28 |

| CKSEL1 | Counter 1 Clock Source |
|---|---|
| 0 | 10 MHz clock |
| 1 | 1 MHz clock |

| CKSEL2 | Counter 2 Clock Source |
|---|---|
| 0 | 1 MHz clock |
| 1 | External clock - applied at pin 31 |

## Counter Cascading

Counters may be linked into 32 or 48 bit depth. Counter cascading will override any other counter source selection for counters 1 and 2.

| **CLK1** | **CLK0** | **Counter Cascading** |
|---|---|---|
| 0 | 0 | 3, 16 bit counters |
| 0 | 1 | 1, 16 bit counter (Counter 0) and 1, 32 bit counter (Counter 1 cascaded into Counter 2) |
| 1 | 0 | 1, 48 bit counter (Counter 0 cascaded into Counter 1 and Counter 2) |
| 1 | 1 | Not Defined |

## Counter Gates

The counter gates are tied high through a 10K resistor. In this manner the gates are always enabled. Counters 0 and 2 may be disabled by bringing the gates to ground at pin 27 (Counter Gate 0) and pin 30 (Counter Gate 2). Counter Gate 1 has no external access so counter 1 is always enabled.

## Reading and Loading the Counters

If you attempt to read an active counter, you will most likely get erroneous data. This is partly caused by carries rippling through the counter during the read operation. Also, the low and high bytes are read sequentially rather than simultaneously and, thus, it is possible that carries will be propagated from the low to the high byte during the read cycle. To circumvent these problems, you should perform a counter-latch operation in advance of the read cycle. To do this, load the RW1 and RW2 COUNTER CONTROL BITS with zeroes. This instantly latches the count of the selected counter (selected via the SC1 and SC0 bits) in a 16-bit hold register. A subsequent read operation on the selected counter returns the held value. Latching is the best way to read an active counter without disturbing the counting process. You can only rely on directly-read counter data if the counting process is suspended while reading, by bringing the gate low, or by halting the input pulses.

For each counter you must specify in advance the type of read or write operation that you intend to perform. You have a choice of loading/reading (a) the high byte of the count, or (b) the low byte of the count, or (c) the low byte followed by the high byte.

## Counter Programming Examples

### Using Counter #0 as a Pulse Counter

Note that the counters are “down” counters so, when resetting them, it’s better to load them with a full count value of 65,535 rather than zero.

```
outportb(BASEADDRESS + 7,0x30);       /* counter 0, mode 0 */
outportb(BASEADDRESS + 4,0xff);       /* counter 0 low load byte */
outportb(BASEADDRESS + 4,0xff);       /* counter 0 high load byte */
```

### Reading Counter #0

```
outportb(BASEADDRESS + 7,0x30);       /* counter 0, latch command */
/* read in both bytes of the latched value and combine into an integer */

value = inportb(BASEADDRESS + 4) + (inportb(BASEADDRESS + 4) * 256;
```

### Generating Square Waves of Programmed Frequency

Frequency of output is a direct function of the frequency of the clock input and of the count loaded into the counter. The minimum count (or divisor) is 2 and the maximum is 65535.

Calculating what divisor to use for a specific output frequency is straightforward. If, for example, you desire a 2000 Hz output and your clock source is 10 MHz, divide it by 2000 and find that the count loaded into counter #0 should be 5000.

## Measuring Frequency and Period

The two previous sections show how to count pulses and generate output frequencies. It is also possible to measure frequency by raising the gate input of Counter #0 for a known time interval and counting the number of clock pulses accumulated for that interval. The gating signal can be derived from Counters #1 and #2 operating in a square wave mode.

Counter #0 can also be used to measure pulse width or half period of a periodic signal. The signal should be applied to the gate input of Counter #0 and a known frequency applied to the Counter #0 clock input. During the interval when the gate input is low, Counter #0 is loaded with a full count of 65,535. When the gate input goes high, the counter begins decrementing until the gate input goes back low at the end of the pulse. The counter is then read and the change in counts is a linear function of the duration of the gate input signal. If Counter #0 receives 10 microsecond duration clock pulses (100 KHz), the maximum pulse duration that can be measured is $65{,}535*10^{-5}$ = 655 milliseconds.

## Generating Time Delays

There are four methods of using Counter #0 to generate programmable time delays.

### Pulse on Terminal Count

After loading, the counter output goes low. Counting is enabled when the gate goes high. The counter output will remain low until the count reaches zero, at which time the counter output goes high. The output will remain high until the counter is reloaded by a programmed command. If the gate goes low during countdown, counting will be disabled as long as the gate input is low.

### Programmable One-shot

The counter need only be loaded once. The time delay is initiated when the gate input goes high. At this point the counter output goes low. If the gate input goes low, counting continues but a new cycle will be initiated if the gate input goes high again before the timeout delay has expired; i.e., is re-triggerable. At the end of the timeout, the counter reaches zero and the counter output goes high. That output will remain high until re-triggered by the gate input.

### Software Triggered Strobe

This is similar to Pulse-on-Terminal-Count except that, after loading, the output goes high and only goes low for one clock period upon timeout. Thus, a negative strobe pulse is generated a programmed duration after the counter is loaded.

### Hardware Triggered Strobe

This is similar to Programmable-One-Shot except that when the counter is triggered by the gate going high, the counter output immediately goes high, then goes low for one clock period at timeout, producing a negative-going strobe pulse. The timeout is re-triggerable; i.e., a new cycle will commence if the gate goes high before a current cycle has timed out.

This page intentionally left blank

# Chapter 7: Specifications

**Features**

24 channels of unbuffered digital input/output.
Four and Eight bit groups independently selectable for I/O
Interrupt and Interrupt-disable capability.
External and internal interrupt and counter source capacity.
Three 16-bit counter/timers.

**Digital Inputs/Outputs**

| | |
|---|---|
| Logic High | 2.0 to 7.0 VDC. |
| Logic Low | -0.3 to 0.8 VDC. |
| Input Load (Hi) | +3.2 microamperes. |
| Input Load (Lo) | -250 microamperes. |
| Max Drive | 5 LSTTL Loads |

**Counters:**

| | |
|---|---|
| Number | Three |
| Type | 16 bit, 65535 counts |
| Frequency | 10 MHz Max, Internal or external source |
| Output High | 2.4 V Min @ -250 mA |
| Output Low | 2.0 V Min, 7.3 Max |

**Interrupt:**

| | |
|---|---|
| Level | 2-6, 7, 10-15 set by CFG configuration file |
| Enable | Programmable |
| Trigger | External or Internal |

**Environmental:**

| | |
|---|---|
| Operating Temperature Range | 0 to +60 ℃ |
| Storage Temperature Range | -50 to +120 °C |
| Humidity | 5% to 95%, non-condensing. |
| Power Required | 5 VDC, 47 mA Typical, 65 mA Max |
| Size | Type II PCMCIA card slot. |

**Regulatory Compliance:**

FCC Part 15, Level A: Designed to meet.
CE Certification not completed at print time.
Please check with us for certification status

# 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: **Model PIOD24**

Manual Revision: **00650-141-1A**

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