# MODEL WDG-CSM

# USER MANUAL

FILE: wdg-csm

# Notice

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

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

# Warranty

Prior to shipment, ACCES equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, ACCES assures its customers that prompt service and support will be available. All equipment originally manufactured by ACCES 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 ACCES' 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 ACCES 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 ACCES option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, ACCES 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 ACCES

Equipment provided but not manufactured by ACCES 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 ACCES is limited to replacing, repairing or issuing credit (at ACCES discretion) for any products which are proved to be defective during the warranty period. In no case is ACCES 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 ACCES equipment not approved in writing by ACCES or, if in ACCES 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 ACCES.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1: Introduction

This multifunction card contains a watchdog timer. Additionally, one or more options may be included on your card. These are: a computer power supply monitor, a computer internal temperature monitor, an optional capability to read the temperature, a fan-speed monitor, and an on-card alarm to signal watchdog timeout.

## Watchdog

Your application program must communicate with the watchdog circuit at prescribed intervals. If this communication ("prompt") is missed, the Watchdog can be programmed to initiate a computer reset (reboot). If the reboot is successful, operation, may be returned to the previous application program. If the failure was temporary, proper operation is resumed. If, however, the failure is persistent, the Watchdog will continuously reset the computer. The more frequently the Watchdog is prompted (and shorter Watchdog time selected), the less time a faulty computer has to cause damage.

The method used by the WDG-CSM card to detect loss of computer function is as follows:

A type 82C54 counter/timer is used. This chip contains three 16-bit counters. A number greater than zero is set into the chip's Counters 0 and 1 by your application program. The Watchdog is armed by software command and both counters begin counting down. As long as the computer is operating properly, both counters will be periodically reloaded to their original programmed values by your application program before both counters have counted down to zero.

If your software fails to reload the counters, then both counters continue counting until zero is reached (timeout). When the counters 0 and 1 reach zero, either the power-good line is held low for approximately 16 milliseconds (performing a hardware reset) through a relay contact, or external lines (either relay contacts or a de-bounced opto-isolated switch) are active while an alarm sounds (if enabled). When a reset condition occurs, the reset circuit is active until a reset pulse returns from the system bus or power is cycled to the system.

The clock frequency to Counter 0 is derived from the computer's color clock and is 894.88625 KHz. (The period is 1.117 $\mu$sec.) The output of Counter 0 is used as a clock to Counter 1. Since each counter can divide by any whole number from 2 to 65,536 ($2^{16}$), the watchdog timeout period may vary from about 4 microseconds to 4800 seconds.

The Watchdog card can generate an interrupt request one Counter 0 period-width before the reset timeout. For example, if a reset period of 60 seconds is used with a 5 millisecond delay stored in Counter 0 (the result of a maximum value delay), an interrupt would occur at 59.995 seconds. This gives the Interrupt handler software 5 milliseconds to refresh the watchdog before a reset action occurs. This should allow your software to take corrective actions if the system software continued to run but the software that should have reset the watchdog had failed. Discussion of these Watchdog Programming Options starts on page 5-1.

The interrupt request (IRQ) output is tri-stated at a high-impedance when it is not sending an interrupt request (1 μsec). Thus, that IRQ number can be shared with other I/O cards that have shareable ability. IRQs 2 through 7, 10, 11, 12, 14, and 15 are available.

There are several outputs from the watchdog circuit: (See Chapter 6 for pinout.)

a.      Double-pole double-throw, Form C, relay contacts on the rear panel I/O connector.
b.      An opto-isolated reset output on the rear panel I/O connector.
c.      An opto-isolated complement of  the reset output on the rear panel I/O connector.
d.      A FAN IN input on the rear panel I/O connector.
e.      TTL reset signal on internal terminal block TB1.
f.      The complement of that output at terminal 3 of TB1.
g.      A Watchdog 56KHz. pulse on the rear panel I/O connector.
h.      Un-fused 5V DC output.

As noted in items b. and c. above,  opto-coupler outputs (one ON when the other is OFF) are provided for use where relay contact bounce could be a problem. Further, as noted in e. above, a buffered discrete output is also provided. This output goes high to signal a watchdog reset condition.  Finally, a 56 KHz., TTL-level, 50 percent duty cycle signal is provided at I/O connector pin 13 when the watchdog circuit is enabled and no reset is in progress. Otherwise, this output is in a low state.

## Options

Your card may have one or more options installed as mentioned in the opening paragraph of this description. The following paragraphs describe these options.

### Option 01:  Computer Power Monitor

The four computer power supplies (+5V, -5V, +12V, and -12V) are monitored. If one or more of those voltages are more than ±6 percent outside of their nominal values, then two bits of the Status Register indicate whether there is an overvoltage or an undervoltage. In addition, an interrupt request can be generated.

### Option 02:  Computer Temperature Monitor

If this option is installed, Option 01 must also be installed. This option monitors ambient temperature inside the computer chassis. The temperature monitor circuit compares the output of an LM334 temperature sensor with a preset DC voltage level. The output of the comparator circuit can be read at a bit location of the Status Register and, also, can cause an interrupt request if that temperature exceeds the factory preset limit (50 °C).

### Option 03:  Computer Temperature Measurement

This option requires presence of both Option 01 and 02. When this option is included, an onboard 8 bit A/D converter provides means for a software read of the measured temperature. Resolution is to approx. 0.7 °F.

### Option 04:  This Option Provides Four Functions as Follows:

#### Change of State

Differential digital inputs are accepted through pins 17 & 18 (ISOIN0) and pins 19 & 20 (ISOIN1), are opto-isolated and reported in the Status Register. The change-of-state also generates an IRQ interrupt request.

#### Fan Speed

This function is usable only in computers which use fans that have a tachometer output. If fan speed falls to unsafe levels (i.e., if the tachometer output falls to less than 50pps), an interrupt request is generated.

#### Buzzer

The buzzer is under software control and can be turned on by programming a "write" to Base Address + 4 or off by programming a "write" to Base Address +5 if Counter #2 is configured in mode 1. Configuring Counter #2 in mode 0 will defeat the buzzer entirely. (see Programming, Chapter 5, for more information)

#### Opto-Isolated Outputs

This option provides an opto-isolated reset signal at pins 4 and 5 (Isolated Reset Output) of the DB25 connector. An opto-isolated inverse of the reset signal is also provided across pins 6 and 7 (Isolated NOT Reset Output) of the same connector.

### Special Options

Programmable array logic is available on this card and many possibilities exist for special modifications to suit unique requirements. If your card includes any such modifications, there will be an "Addendum" sheet inserted at the front of this manual and an "S" number (e.g. S01, S02, etc) will be appended to the model number.

## Status Register

Bit assignments of the Status Register are as follows:

BD0    Watchdog counter refresh reminder (Active low)
BD1    Temperature good (Active high)
BD2    Isolated Input #1 status (Same as input)
BD3    Isolated Input #0 status (Same as input)
BD4    Fan good (Active low)
BD5    Power Supply overvoltage (Active low)
BD6    Power Supply undervoltage (Active low)
BD7    IRQ generated (Active low)

## Utility Software

Utility software are provided on the CD with the WDG-CSM card. This software includes:

a.    A base-address locator program, FINDBASE, to help you select a base address that will not conflict with other installed computer resources.
b.    SETUP.EXE: an illustrated setup program.
c.    Sample programs for use with QuickBASIC, Turbo-C and Turbo-Pascal.  Sample 1 demonstrates how to program the counter/timer and Sample 2 displays each bit of the Status Register and shows the measured temperature inside the computer.
d.    Programming hints are provided in Chapter 5, Programming of this manual.

## Specifications

### Watchdog Timer

- Time-out:      Software selectable from 4 μsec. to 4800 seconds.

- Output Pulse Width:      16 msec. minimum.

- Clock:      894.88625 KHz, derived from color clock (14.31818MHz÷16).

- Address:      Continuously mappable within 000 to 3FF hex I/O range.

- Relay Output:      DPDT, Form C, max. switching current 2A DCat 30W or 1.25A, AC at 30VA max.

- Interrupt Output:      Jumper selectable, IRQ 2-7, 10, 11, 12, 14, and 15.

### Status Monitor Options

- Voltages   IRQ and status register indication if +5, -5, +12, or -12 V exceed +/- 6% of nominal.

- Temp. Alarm:      IRQ and status register indication at 122 °F and above.

- Temp. Sensor:      8 bit ADC, LSB = 0.7°F (factory adjustable).

- Fan Speed:      IRQ and status register indication whenever tachometer output drops below 50pps (custom factory settings available).

- Buzzer:      Audio Alert signals watchdog timeout.

- LED Output:      5V, through 470Ω resistor.

- Isolated Outputs:      Complementary opto-isolated reset outputs (2).

- Isolated Inputs:      Two, opto-isolated, 5 mA typical load. Can be read at base address + 4.

- Power Required:      +5 VDC at 125 mA with no options, 250 mA with all options installed.

- Size:      6.5 inches long (165 mm) x 3.9 inches high (99 mm).

### Environmental

- Operating Temperature Range:   0 °C. to +60 °C.

- Storage Temperature Range:   -50 °C. to +120 °C.
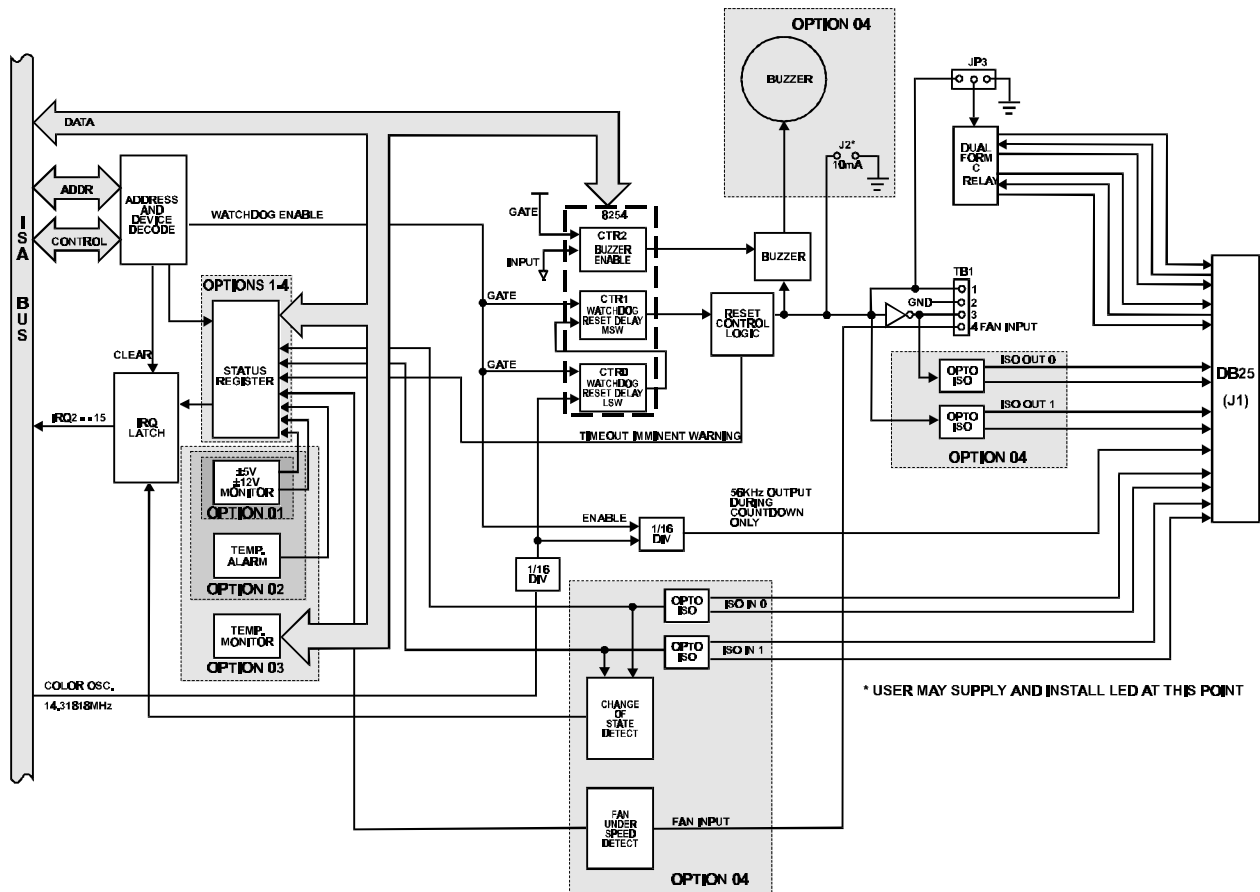
- Humidity:      10% to 90% RH, non-condensing.

**Figure 1-1:** WDG-CSM Block Diagram

# Chapter 2:  Installation

The software provided with this card is contained on either one CD or multiple diskettes  and must be installed onto your hard disk prior to use.  To do this, perform the following steps as appropriate for your software format and operating system. Substitute the appropriate drive letter for your CD-ROM or disk drive where you see  d: or  a: respectively in the examples below.

## CD Installation

### DOS/WIN3.x
   a.  Place the CD into your CD-ROM drive.
   b.  Type d:K to change the active drive to the CD-ROM drive.
   c.  Type installK to run the install program.
   d.  Follow the on-screen prompts to install the software for this card.

### WIN95/98/NT
   a.  Place the CD into your CD-ROM drive.
   b.  The CD should automatically run the install program after 30 seconds.  If the install program does not run, click START | RUN and type d:install, click OK or press K.
   c.  Follow the on-screen prompts to install the software for this card.

## 3.5-Inch Diskette Installation

As with any software package, you should make backup copies for everyday use and store your original master diskettes in a safe location. The easiest way to make a backup copy is to use the DOS DISKCOPY utility.

In a single-drive system, the command is:

diskcopy a: a:K

You will need to swap disks as requested by the system.
In a two-disk system, the command is:

diskcopy a: b:K

This will copy the contents of the master disk in drive A to the backup disk in drive B.

To copy the files on the master diskette to your hard disk, perform the following steps.
1.     Place the master diskette into a floppy drive.
2.     Change the active drive to the drive that has the diskette installed. For example, if the diskette is in drive A, type a:K.
3.     Type  installK and follow the on-screen prompts.

## Directories Created on the Hard Disk

The installation process will create several directories on your hard disk.  If you accept the installation defaults, the following structure will exist.

### [CARDNAME]
Root or base directory containing the SETUP.EXE setup program used to help you configure jumpers and calibrate the card.

**DOS\PSAMPLES:**          A subdirectory of  [CARDNAME] that contains Pascal samples.
**DOS\CSAMPLES:**          A subdirectory of [CARDNAME] that contains "C" samples.
**Win32\language:**          Subdirectories containing samples for Win95/98 and NT.

### WinRisc.exe
A Windows dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line.  Can be used to say hello to an installed modem.

### ACCES32
This directory contains the Windows 95/98/NT driver used to provide access to the hardware registers when writing 32-bit Windows software.  Several samples are provided in a variety of languages to demonstrate how to use this driver. The DLL provides four functions (InPortB, OutPortB, InPort, and OutPort) to access the hardware.

This directory also contains the device driver for Windows NT, ACCESNT.SYS.  This device driver provides register-level hardware access in Windows NT.  Two methods of using the driver are available, through ACCES32.DLL (recommended) and through the DeviceIOControl handles provided by ACCESNT.SYS (slightly faster).

### SAMPLES

Samples for using ACCES32.DLL are provided in this directory. Using this DLL not only makes the hardware programming easier (MUCH easier), but also one source file can be used for both Windows 95/98 and WindowsNT. One executable can run under both operating systems and still have full access to the hardware registers. The DLL is used exactly like any other DLL, so it is compatible with any language capable of using 32-bit DLLs. Consult the manuals provided with your language's compiler for information on using DLLs in your specific environment.

### VBACCES

This directory contains sixteen-bit DLL drivers for use with VisualBASIC 3.0 and Windows 3.1 only. These drivers provide four functions, similar to the ACCES32.DLL. However, this DLL is only compatible with 16-bit executables. Migration from 16-bit to 32-bit is simplified because of the similarity between VBACCES and ACCES32.

### PCI

This directory contains PCI-bus specific programs and information. If you are not using a PCI card, this directory will not be installed.

### SOURCE

A utility program is provided with source code you can use to determine allocated resources at run-time from your own programs in DOS.

### PCIFind.exe

A utility for DOS and Windows to determine what base addresses and IRQs are allocated to installed PCI cards. This program runs two versions, depending on the operating system. Windows 95/98/NT displays a GUI interface, and modifies the registry. When run from DOS or Windows3.x, a text interface is used. For information about the format of the registry key, consult the card-specific samples provided with the hardware. In Windows NT, NTioPCI.SYS runs each time the computer is booted, thereby refreshing the registry as PCI hardware is added or removed. In Windows 95/98/NT PCIFind.EXE places itself in the boot-sequence of the OS to refresh the registry on each power-up.

This program also provides some COM configuration when used with PCI COM ports. Specifically, it will configure compatible COM cards for IRQ sharing and multiple port issues.

### WIN32IRQ

This directory provides a generic interface for IRQ handling in Windows 95/98/NT. Source code is provided for the driver, greatly simplifying the creation of custom drivers for specific needs. Samples are provided to demonstrate the use of the generic driver. Note that the use of IRQs in near-real-time data acquisition programs requires multi-threaded application programming techniques and must be considered an intermediate to advanced programming topic. Delphi, C++ Builder, and Visual C++ samples are provided.

### Findbase.exe

DOS utility to determine an available base address for ISA bus , non-Plug-n-Play cards.  Run this program once, before the hardware is installed in the computer, to determine an available address to give the card.  Once the address has been determined, run the setup program provided with the hardware to see instructions on setting the address switch and various option selections.

### Poly.exe

A generic utility to convert a table of data into an nth order polynomial.  Useful for calculating linearization polynomial coefficients for thermocouples and other non-linear sensors.

### Risc.bat

A batch file demonstrating the command line parameters of RISCTerm.exe.

### RISCTerm.exe

A dumb-terminal type communication program designed for RS422/485 operation.  Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line.  Can be used to say hello to an installed modem.  RISCTerm stands for Really Incredibly Simple Communications TERMinal.

## Installing the Card

Before installing the WDG-CSM card, carefully read Chapter 3 and Chapter 4 of this manual and configure the card according to your requirements. The SETUP Program can be used to assist in configuring jumpers and switches on the card. Be especially careful with Address Selection. If the addresses of two installed functions overlap, you will experience unpredictable computer behavior. To help avoid this problem, refer to the FINDBASE.EXE program provided with your card.

### To Install the Card

1.   Remove power from the computer.
2.   Remove the computer cover.
3.   Remove the blank I/O backplate.
4.   Install jumpers for selected options. See the WDGSET utility program and Chapter 3 of this manual, Option Selection.
5.   Select base address.  See the FINDBASE utility program and Chapter 4 of this manual, Address Selection.
6.   Install the card in an I/O expansion slot. Make sure that the card mounting bracket is properly screwed into place and that there is a positive chassis ground.
7.   Install the "Reset" wire from TB1-1 to the "Power Good" signal on the power supply connector J8, pin 1 using the clamp provided or to the active terminal of the Reset switch.
8.   Inspect for proper fit of the card and connectors and tighten screws.
9.   Replace the computer cover.

## Connecting the Reset Line

For an ATX type power supply, you must connect to the hot line (+5VDC) of the reset push button wires. For AT-type supplies, you can use either the power good line on P8, or the hot side of the reset push button. You can determine which reset push button wire is hot by removing the reset pushbutton connector from the motherboard and measuring for +5VDC on the pins. Alternatively, you can try to connect into one wire, running the sample program, and if it doesn't work, try connecting to the other wire. To make a connection, insert the power-good or reset push button line into the clamp, compress the metal bar with pliers and close the protective cover.

POWER GOOD LINE,
PIN ONE (1),
USUALLY ORANGE

To WATCHDOG TERMINAL BLOCK

P8

P9
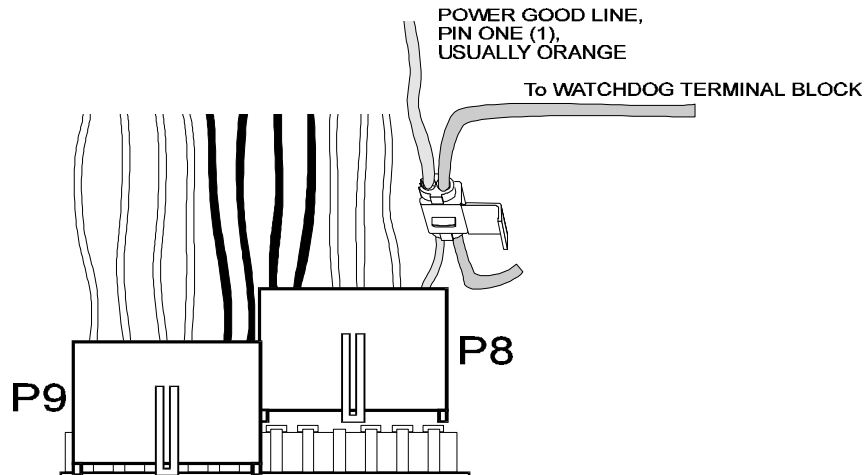
# Chapter 3:  Option Selection

Refer to Figure 1-1, Watchdog Block Diagram  and Figure 3-1, Option Selection Map  in this chapter when reading this section of the manual.  Card operation is determined by jumper installation as described in the following paragraphs.

## Interrupts

Interrupts are enabled by installing a jumper at locations marked IRQ2, 3, 4, 5, 6, 7 (at jumper block JP1) or marked IRQ 10, 11, 12, 14, and 15 (at jumper block JP2).

## Relay Enable/Disable

Hold the card with the fingers at the bottom and the mounting bracket on the right. Jumper block JP3 is located immediately above the relay at the right-hand side of the card. If no jumper is placed on the block, the relay is always de-energized. If a jumper is placed between the two right-hand posts, the relay is always energized when +5V power is on. And, if a jumper is placed between the two left-hand posts, the relay is energized except during reset (watchdog timeout). If you desire to use the relay output, then the jumper should be placed between the left-hand posts.

## Screw Terminals

Screw terminals located on TB1 at the top left-hand side of the card provide the means for connections of the watchdog output to points inside the computer chassis. Terminal 1 provides the active-low watchdog output. Terminal 2 provides a ground. Terminal 3 provides the complementary active-high of the Terminal 1 output.  Terminal 4 provides an  input for the output of the fan motor tachometer or fan-good sensor.

## LED Terminals

A connector for an external or on-board LED is provided at two solder pads labeled J2. The output is limited by a 470$\Omega$ resistor in series with a 5V output. This output is only active if the Buzzer option is installed.
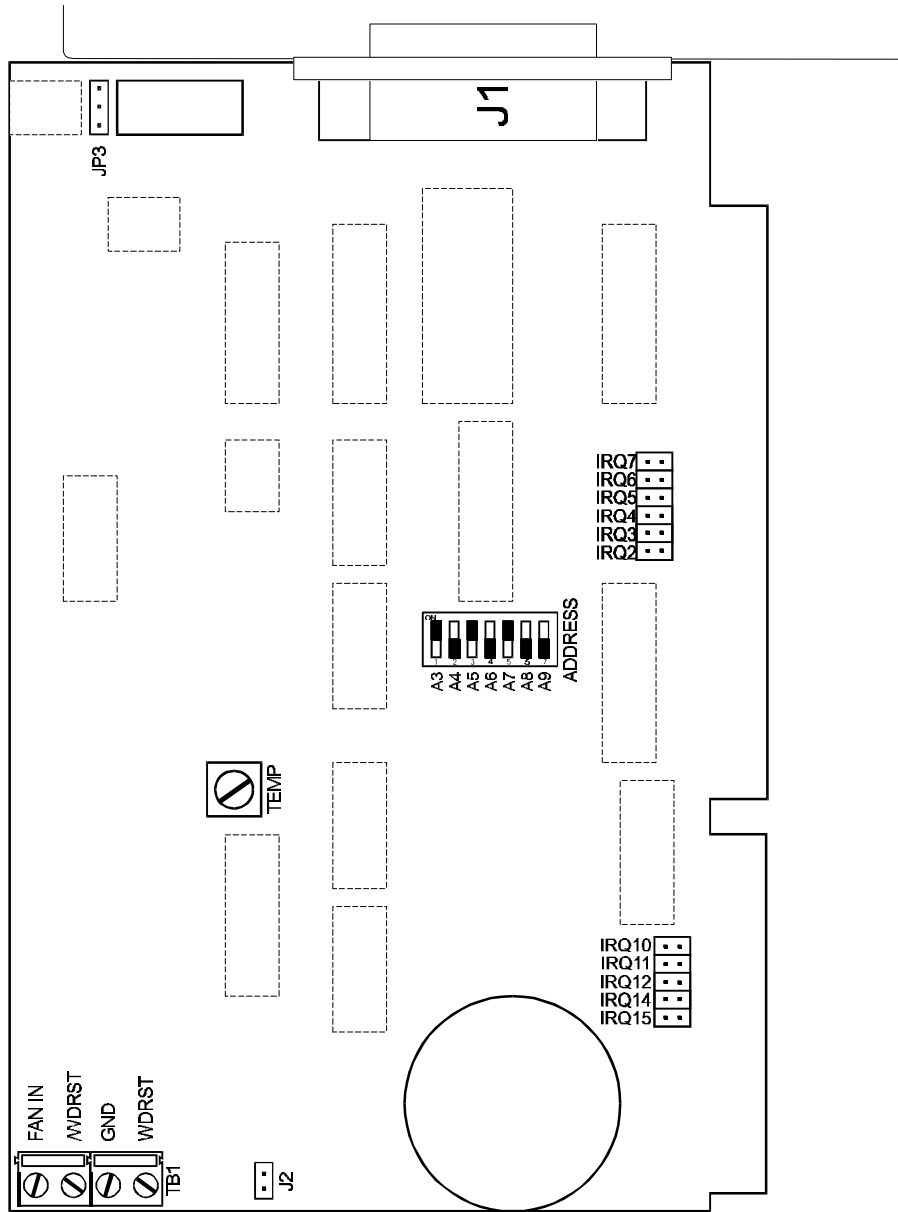
**Figure 3-1:** WDG-CSM Option Selection Map

# Chapter 4: Address Selection

The card base address on the I/O bus is set by DIP switch S1. The switches are marked A3 through A9 and A3 is the least significant bit of the address. The base addresses can be selected anywhere within the I/O address range 000-3FF provided that they do not overlap with other functions. The FINDBASE software utility provided on CD with your card will help you select a base address that does not conflict with other assignments. If in doubt, refer to the following table for a list of standard address assignments.

| Hex Range | Usage |
|-----------|-------|
| 000-01F | DMA Controller 1 |
| 020-03F | INT Controller 1, Master |
| 040-05F | Timer |
| 060-06F | 8042 (Keyboard) |
| 070-07F | Real Time Clock, NMI Mask |
| 080-09F | DMA Page Register |
| 0A0-0BF | INT Controller 2 |
| 0C0-0DF | DMA Controller 2 |
| 0F0 | Clear Math Coprocessor Busy |
| 0F1 | Reset Coprocessor |
| 0F8-0FF | Arithmetic Processor |
| 1F0-1F8 | Fixed Disk |
| 200-207 | Game I/O |
| 278-27F | Parallel Printer Port 2 |
| 2F8-2FF | Asynchronous Comm'n (Secondary) |
| 300-31F | Prototype Card |
| 360-36F | Reserved |
| 378-37F | Parallel Printer Port 1 |
| 380-38F | SDLC or Binary Synchronous Comm'n 2 |
| 3A0-3AF | Binary Synchronous Comm'n 1 |
| 3B0-3BF | Monochrome Display/Printer |
| 3C0-3CE | Local Area Network |
| 3D0-3DF | Color/Graphic Monitor |
| 3F0-3F7 | Floppy Diskette Controller |
| 3F8-3FF | Asynchronous Comm'n (Primary) |

**Table 4-1:** Standard Address Assignments for 286/386/486 Computers

Address Setup switch locations are marked A3 through A9.

In order to configure the desired address, the hexadecimal address must be converted to a binary representation.

For example, as illustrated below, switch selection corresponds to hex 2D8 (or binary 10 1011 1xxx). The "xxx" represents address lines A2, A1, and A0 used on the card to select individual registers as described in the Chapter 5, Programming of the manual.

| Hex Representation | 2 | | D | | | | 8 |
|---|---|---|---|---|---|---|---|
| Conversion Factors | 2 | 1 | 8 | 4 | 2 | 1 | 8 |
| Binary Representation | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| Switch Setting | Off | On | Off | Off | On | Off | Off |
| Switch Label | A9 | A8 | A7 | A6 | A5 | A4 | A3 |

Please note that "1" = "off" and that "0" = "on."

Review the Address Selection Table carefully before selecting the card address. If the addresses of two installed functions overlap, you will experience unpredictable computer behavior. If you have doubts concerning available addresses in your particular computer, use the FINDBASE utility provided to determine available addresses.

# Chapter 5: Programming

This section of the manual contains information to assist you in developing programs for use with the card. I/O bus address assignments, programming hints, a program example, and a description of the VisualBASIC utility driver are included.

The program example provided is intended as a guide rather than working software. ACCES assumes no liability for it's use.

## Monitor Functions

The WDG-CSM card uses eight consecutive addresses in I/O space as listed in the following table.

| Address | Read | Write |
|---------|------|-------|
| Base Address | Read Counter #0 | Write to Counter #0 |
| Base Address +1 | Read Counter #1 | Write to Counter #1 |
| Base Address +2 | Read Counter #2 | Write to Counter #2 |
| Base Address +3 | Read Control Register | Write to Control Register |
| Base Address +4 | Read Status Register | Start Buzzer (if enabled)† |
| Base Address +5 | Read Temperature | Stop Buzzer † |
| Base Address +7 | Disable Counters | Enable Counters |

Buzzer operation requires configuring Counter #2 in mode 1

**Table 5-1:** Register Address Map

### Computer Temperature Monitor

If Option 03 is installed, the temperature inside the computer can be read using an 8-bit register at BASE+5. To convert the byte read into temperature in Fahrenheit, multiply the value read by 11/15 and then add 7.

## Interrupts

The card is capable of generating interrupts for a variety of reasons. If any or all of the other standard options described on page 2-3 are installed (as well as the watchdog-timeout-near warning), they can be read from the Status Register at BASE+4. Bit assignments are as listed below. An interrupt is also generated upon a digital input's change-of-state if that option is installed.

## Status Register

Bit assignments of the Status Register (Base Address +4) are as follows:

BD0    Watchdog counter refresh reminder (Active low).
BD1    Temperature good (Active high).
BD2    Isolated Input #1 status (Same as input).
BD3    Isolated Input #0 status (Same as input).
BD4    Fan good (Active low).
BD5    Power Supply overvoltage (Active low).
BD6    Power Supply undervoltage (Active low).
BD7    IRQ generated (Active low).

## Watchdog Programming Options

Programming WDG-CSM is straightforward. Counters 0 and 1 of the 8254 counter/timer are concatenated and operate as a 32-bit down counter to provide the time delay before a timeout occurs. A special register provides enable and disable functions for the counters and, thus, the watchdog itself. Writing any value to the register located at base address +7 enables the counters to count. Reading the register disables the counters and stops the watchdog.

### To program the watchdog interval
1.    Disable the counters.
2.    Program the counters with the desired time delays.
3.    Enable the counters.

Once the counters have been enabled, the computer will reset if the 32-bit counter decrements all the way to zero. (Note: An alternative use of the reset signal is to simply notify an external device that a reset is necessary.)

### Foreground Watchdog Mode
In order to prevent the reset, the software program must periodically reload Counter 1 with the initial load value (or any other suitable load value). The maximum delay between Counter 1 reloads is determined by the load values of the concatenated counters minus a suitable fudge factor.

### Note

It is not necessary to re-load Counter 0 because it will simply recycle. Also note that loading Counter 0 with full-scale is customary, as exact timing is rarely a requirement of a watchdog function.

Counter 0 must be programmed for mode 3. Counter 1 must be programmed for mode 2. The WDG-CSM was designed specifically for these modes. Any other modes may cause unpredictable behavior. See Appendix A for details on how to program the counters.

In Summary:

| | | |
|---|---|---|
| 1. | Disable Watchdog | Read BASE+7 |
| 2. | Program CTR0 for Mode 3 | See Appendix A |
| 3. | Program CTR1 for Mode 2 | See Appendix A |
| 4. | Load CTR0,1 with reset delay | See Appendix A |
| 5. | Enable Watchdog | Write BASE+7 |

Then, before the first timeout occurs:

| | | |
|---|---|---|
| 1. | Enter main loop | |
| 2. | Program CTR1 (again) for Mode 2 | |
| 3. | Load CTR1 | See Appendix A |
| 4. | User code goes here | |
| 5. | Repeat main program loop | |

## Background Watchdog Mode

An alternative method to refresh the counters that provides some flexibility but potentially reduces dependability is to use an interrupt service routine (ISR). WDG-CSM will generate an interrupt one Counter 0 load value before timeout occurs. If that interrupt is used by a user-defined ISR, it's possible for that ISR to refresh the counters and, thus, avoid need for the main program loop to refresh the counters.

There is possibility of reduced dependability because this method will not detect certain types of program lock-ups. For example, if the application program were to lock up in a loop such as "JMP $", the interrupt service routine would never know it and would therefore continue refreshes despite the lockup. It is possible to modify the host program to enable the ISR to detect such loops (perhaps by checking the return stack location for the ISR each time, and making sure it changes...) However, if the code is going to be modified, the next, hybrid, method might be more effective.

This method does, however, allow "off-the-shelf" existing programs to be used with WDG-CSM without modification. Also, this method could cause spurious resets if your program disables interrupts for an extended duration. The IRQ that would have caused the ISR might be missed and allow the counter to time out. If the counter counts-down, the reset signals will be generated and the counters will restart.

## Combo Reset Mode

By combining the two methods ("Foreground" and "Background") it is possible to create a very elaborate scheme for monitoring the status of the computer. This combined method eliminates the negatives associated with either method used individually, but does require extensive modifications to existing code, or even rewriting code entirely. In this scheme, both foreground and background routines can prompt the watchdog.

The foreground routine would prompt the watchdog under normal conditions, and the ISR running in the background would only prompt the watchdog if the foreground routine failed to do so.  The ISR, noting that the foreground routine missed it's prompt, could take steps to determine why, and even try to correct it. For example, the foreground program can post it's current status to a shared-memory variable, indicating that it is about to enter a long dedicated process (calculations for a print job, perhaps) and that it might miss several of its prompts. The ISR, when executed by the warning IRQ, notes that the foreground program indicated it might miss its turn, and starts counting misses. After prompting the watchdog, it returns control to the foreground process. If the count of missed prompts gets too high, it could write status to a disk file or serial port, then skip the prompt, allowing the computer to reset.

## Sample Program

The following program is written in Pascal. This program will set up Counter 0 with full scale (65,535) and Counter 1 with 100 and send out a reset pulse when reaching zero.

```
{
================================================================
            PASCAL Language Sample #1: SAMPLE.PAS

 This program will set up counter 1 with 100, and counter 0
 with full scale (65535) sending out a pulse at zero.
================================================================
}
USES
  Crt;

CONST
  ADDRESS : WORD = $350; {The Base address of the watchdog
board.}

    procedure CtrMode(cntr,mode:byte);
    var ctrl:byte;
    begin
     ctrl := (cntr shl 6) or $30 or (mode shl 1);
     port[ADDRESS+3] := ctrl;
    end;

    procedure LoadCtr(c,val:integer);
    begin
     port[ADDRESS+c] := lo(val);
     port[ADDRESS+c] := hi(val);
    end;
    function ReadCtr(c:integer) : integer;
    begin
     port[ADDRESS+3] := c shl 6;
     ReadCtr := port[ADDRESS+c] + (port[ADDRESS+c] shl 8);
```

```
      end;

{
*************************************************************
PROCEDURE set_counter
Parameters:  None
Returns:     None

Purpose: This procedure performs the setup operations on the
counter/timer chip.
Calls:    None
Uses:     ADDRESS          Constant defined at top of program.
*************************************************************
}
PROCEDURE set_counter;
VAR
  temp,control_word : BYTE;
BEGIN
  temp:=port[ADDRESS+7];
  CtrMode(0,3);     {program the counters for the REQUIRED modes}
  CtrMode(1,2);
  CtrMode(2,1);
  LoadCtr(0,-1);   {-1 is full load value, long reset and high
                    granular}
  LoadCtr(1,10);
 END; { End set_counter }

VAR
  read_back : BYTE;
  loop : WORD;
  ch : CHAR;
BEGIN
  clrscr;
  writeln('Pascal Sample #1: Use Of Counter/Timer Chip with
Watchdog');
  writeln('This program demonstrates how to program the 82C54
counter/timer');
  writeln('chip on the Watchdog Timer board.  The Watchdog
address should be set');
  writeln('to 350 hex.  A keystroke will stop the program.');
  writeln;
  writeln('Press any key to start.');
  ch := readkey;                              { Grab keystroke. }

  set_counter;          { programs counters }

  port[ADDRESS+7] := 0; { Starts counters counting }
  for loop := 1 to 1000 do
    begin
      CtrMode(1,2);
```

```
      LoadCtr(1,10);      {prompt counter 1--don't need to prompt 0}
      Writeln('Updating counter 1.  Loop number : ', loop);
      GotoXY(1, 7);
    end;

  writeln;
  writeln('Waiting for timeout . . .');

  repeat
    read_back := port[ADDRESS+4];  { load control value }
  until ((keypressed) OR ((read_back AND 128) = 0));  { check for
bit 0 }

  writeln;
  writeln('Watchdog timed out successfully.');
END.
```

# Chapter 6: Connector Pin Assignments

| Pin Number | Function |
| --- | --- |
| 1 | Common, Relay Pole 1 |
| 2 | Normally Closed Contact, Pole 1 |
| 3 | Normally Open Contact, Pole 1 |
| 4 | Opto-Isolated Reset Out |
| 5 | Opto-Isolated Reset Source |
| 6 | Opto-Isolated NOT Reset Out |
| 7 | Opto-Isolated NOT Reset Source |
| 8 | Counter Enabled |
| 9 | +5 VDC Unfused, 1 A max |
| 10 | +5 VDC " |
| 11 | +5 VDC " |
| 12 | +5 VDC " |
| 13 | 56 KHz Square Wave while WDOG enabled |
| 14 | Common, Relay Pole 2 |
| 15 | Normally Closed Contact, Pole 2 |
| 16 | Normally Open Contact, Pole 2 |
| 17 | Source Opto-Isolated Input #0 |
| 18 | Return Opto-Isolated Input #0 |
| 19 | Return Opto-Isolated Input #1 |
| 20 | Source Opto-Isolated Input #1 |
| 21 | Tachometer Input from Fan Speed Sensor (FAN IN) |
| 22 | Ground |
| 23 | " |
| 24 | " |
| 25 | " |

**Table 6-1:** Rear Panel D Connector Pin Assignments J1

| | |
|---|---|
| **Terminal 1** | TTL Watchdog Output - Active Low |
| **Terminal 2** | Ground |
| **Terminal 3** | TTL Watchdog Output - Active High |
| **Terminal 4** | TTL Input - Fan Speed Sensor (see pin 21 above) (FAN IN) |

**Table 6-2:** On-Board Terminal Block Pin Assignments TB1-1

# Appendix A:  Programmable Interval Timer

This Appendix includes basic information about the type 8254 Counter/Timer chip. For those interested in more detailed information, a full description can be found in the manufacturer's data sheets.

## Operation Modes

Modes of operation are described in the following paragraphs to familiarize you with the power and versatility of this device. The following conventions apply in describing operation of type 8254 chips:

Clock:                          A positive pulse into the counter's clock input.
Trigger:                        A rising edge input to the counter's gate input.
Counter Loading:                Programming 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. This mode is commonly used for event counting with Counter #0.

### 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 retriggerable 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

Like mode 2, this mode operates periodically. 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 waveform.

### 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 go low until the full count after the rising edge of the trigger.

## Programming

On the WDG-CSM card, the 8254 counters occupy the following addresses:

Base Address              Read/Write Counter 0
Base Address  +1          Read/Write Counter 1
Base Address  +2          Read/Write Counter 2
Base Address  +3          Read/Write Counter Control Register

The counters are programmed by writing a control byte into the 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 a follows:

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

SC0 and SC1:  These bits select the counter that the control byte is destined for.

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

RW0 and 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, M1, and 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 (0) or BCD (1).

## Reading and Loading the Counters

If you attempt to read the counters on the fly when there is a high frequency input, 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's possible that "carries" will be propagated from the low byte to the high byte during the read cycle.

To circumvent this, you can perform a counter-latch operation in advance of the read cycle. To do this, load the RW0 and RW1 bits with zeroes. This instantly latches the count of the selected counter in a 16-bit hold register. (An alternative method of latching counters that has an additional advantage of operating simultaneously on several counters is by use of a readback command to be discussed later.) A subsequent read operation on the selected counter returns the held value. Latching is the best way to read a counter on the fly 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.

You must specify in advance the type of read or write operation that you intend to perform for each counter. You have a choice of loading/reading (a) the high byte of the count, (b) the low byte of the count, or (c) the high byte followed by the low byte. This last is of the most general use and is selected for each counter by setting the RW0 and RW1 bits to ones. Of course, subsequent read/load operations must be performed in pairs in this sequence or the sequencing flip-flop in the 8254 chip will get out of step.

The readback command byte format is:

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | CNT | STA | C2 | C1 | C0 | 0 |

CNT:  When is 0, latches the counters selected by bits C0-C2.
STA:  When is 0, returns the status byte of counters selected by C0-C2
C0,C1,C2:  When high, select a particular counter for readback. C0 selects Counter 0, C1 selects counter 1, and C2 selects counter 2.

You can perform two types of readback operations with the readback command. When CNR=0, the counters selected by C0-C2 are latched simultaneously. When STA=0, the counter status byte is read when the counter I/O location is accessed. The counter status byte provides information about the current output state of the selected counter and its configuration. The status byte returned if STA=0 is:

OUT:  Current state of counter output pin.
NC:  Null count. this indicates when the last count loaded into the counter register has actually been loaded into the counter itself. The exact time of load depends on the configuration selected. Until the count is loaded into the counter itself, it cannot be read.

RW0 and RW1:  Read/Write command.
M0-M2:  Counter mode.
BCD:  BCD=0 sets binary mode, otherwise counter is in BCD mode.

If STA and CNT bits in the readback command byte are set low and the RW1 and RW0 bits have both been previously set high in the counter control register (thus selecting two-byte reads), then reading a counter address location will yield:

1st Read:  Status byte
2nd Read:  Low byte of latched data
3rd Read:  High byte of latched data.

After any latching operation of a counter, the contents of its hold register must be read before any subsequent latches of that counter will have any effect. If a status latch command is issued before the hold register is read, then the first read will read the status, not the latched value.

# Customer Comments

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