January 2003

MON08 MULTILINK USER MANUAL

Purchase Agreement

P&E Microcomputer Systems, Inc. reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. P&E Microcomputer Systems, Inc. does not assume any liability arising out of the application or use of any product or circuit described herein.

This software and accompanying documentation are protected by United States Copyright law and also by International Treaty provisions. Any use of this software in violation of copyright law or the terms of this agreement will be prosecuted.

All the software described in this document is copyrighted by P&E Microcomputer Systems, Inc. Copyright notices have been included in the software.

P&E Microcomputer Systems authorizes you to make archival copies of the software and documentation for the sole purpose of back-up and protecting your investment from loss. Under no circumstances may you copy this software or documentation for the purpose of distribution to others. Under no conditions may you remove the copyright notices from this software or documentation.

This software may be used by one person on as many computers as that person uses, provided that the software is never used on two computers at the same time. P&E expects that group programming projects making use of this software will purchase a copy of the software and documentation for each user in the group. Contact P&E for volume discounts and site licensing agreements.

P&E Microcomputer Systems does not assume any liability for the use of this software beyond the original purchase price of the software. In no event will P&E Microcomputer Systems be liable for additional damages, including any lost profits, lost savings or other incidental or consequential damages arising out of the use or inability to use these programs, even if P&E Microcomputer Systems has been advised of the possibility of such damage.

By using this software, you accept the terms of this agreement.

P&E Microcomputer Systems, Inc.
P.O. Box 2044
Woburn, MA 01888
617-353-9206
www.pemicro.com

Manual version 1.03

1 INTRODUCTION

The MON08 MULTILINK is an interface cable whose purpose is to allow debug and programming of 68HC08 devices via the MON08 debug port. The MON08 MULTILINK connects the target to the PC via a standard parallel port.

Some of the features that make the MON08 MULTILINK versatile are:

a. Software configurable port pin settings for Monitor ROM entrance.

b. Works with 2V, 3V, and 5V targets with internal bus frequency ranges from 1MHz to 8MHz.

c. Automatically detects target internal bus frequency and sets communications baud rate.

d. May optionally provide either 2,3, or 5V power @ 125mA to the target via Pin 15 of the MON08 header.

e. Provides 5V 4.9152 MHz oscillator signal to overdrive target crystal and RC clock circuitry.

2 MON08 MULTILINK HARDWARE

2.1 MON08 MULTILINK Power Supply

The MON08 MULTILINK requires a regulated 9V DC Center Positive power supply with 1.3/3.5mm female plug. The MON08 MULTILINK derives its power from the Power Jack located beside the MON08 connector. When the cable is powered up, the Green LED will be on. If the target is powered, the Yellow LED will be on.

2.2 Optional Oscillator

The MON08 MULTILINK provides a 5V 4.9152 MHz oscillator clock signal to Pin 13 of the MON08 Connector. If the target is a 5V system, the user may use this clock signal to overdrive the target RC or crystal circuitry. If this signal is not used, just leave Pin 13 of the target MON08 header unconnected.

Please note that if the target already uses an oscillator as its clock, the MON08 MULTILINK will NOT be able to overdrive it. The clock should have sufficient drive to be used with a target system even if the target system has an RC circuit or crystal connected.

2.3 Target MON08 Connector

The MON08 MULTILINK requires the target to have a standard 16-position 0.100-inch pitch dual row 0.025-inch square header. The mechanical drawing is shown in **Figure 2-1**.

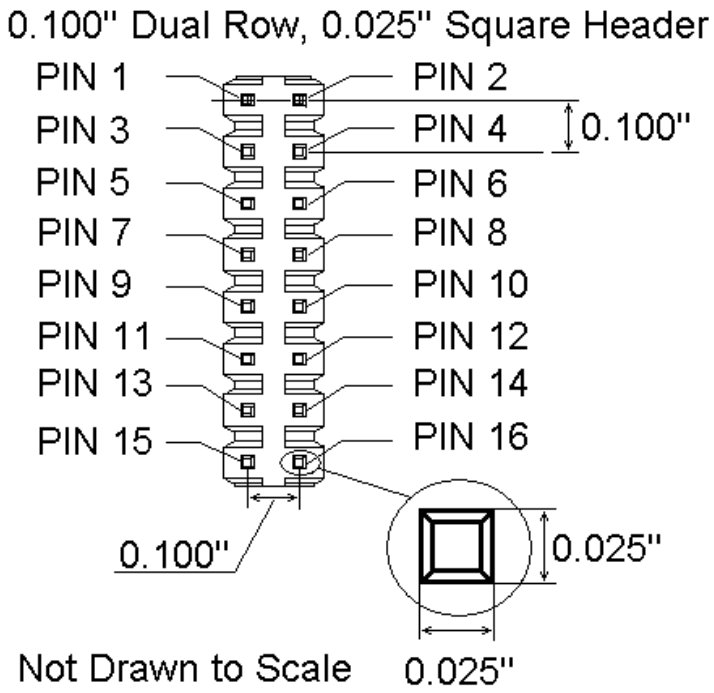

Figure 2-1: 16-Pin Header Mechanical Drawing

The MON08 Connector adopts the standard pin-out from MON08 debugging (as used on different ICS boards) with some modifications. The general pin-out is as follows:

PIN 1 -	**NC**	**GND**	- PIN 2
PIN 3 -	**NC**	**RST**	- PIN 4
PIN 5 -	**NC**	**IRQ**	- PIN 6
PIN 7 -	**NC**	**MON4**	- PIN 8
PIN 9 -	**NC**	**MON5**	- PIN10
PIN11 -	**NC**	**MON6**	- PIN12
PIN13 -	**OSC**	**MON7**	- PIN14
PIN15 -	**Vout**	**MON8**	- PIN16

If viewed right-side-up from the rear (open) end of the cable housing, the MON08 Multilink header looks like this:

Figure 2-2: MON08 Connector Pin Location

Please note that **NC** designates that these pins are reserved for future P&E use. **Make sure you do not connect any signal to these lines**.

The **MON4-MON8** signals are software configurable to support connections to different 68HC908 devices. Depending upon the device, either the MON4 or MON5 pin is the single-wire communications line (which usually corresponds to PORTA0 or PORTB0). The rest of the lines are either no connect or are port lines which must be driven to particular values upon reset. The MON08 MULTILINK software lists the target processor types and their corresponding pin-outs for user references. The software also selects the single-wire communications line according to the target processor type.

2.4 Ribbon Cable

The MON08 MULTILINK communicates with the target through a 16-pin ribbon cable with 0.100-inch centerline dual row socket IDC assembly (not keyed). The ribbon cable is designed such that the MON08 MULTILINK MON08 Connector and the target MON08 Header have the same pinout. i.e. The Pin 1 of the MON08 MULTILINK MON08 Connector is connected to the Pin 1 of the target MON08 Header. **Figure 2-3** sketches the connection mechanism (looking down into the sockets).

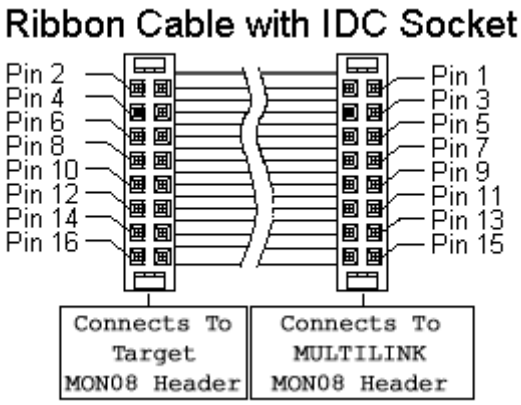

Figure 2-3: Ribbon Cable Diagram

2.5 Target Power Management

The PC software can be configured to have the MON08 MULTILINK drive 2V, 3V, or 5V power to the target on Pin 15 of the MON08 connector. The "device power" option on the connection dialog specifies the voltage level to source. **Figure 2-4** shows the device power settings.

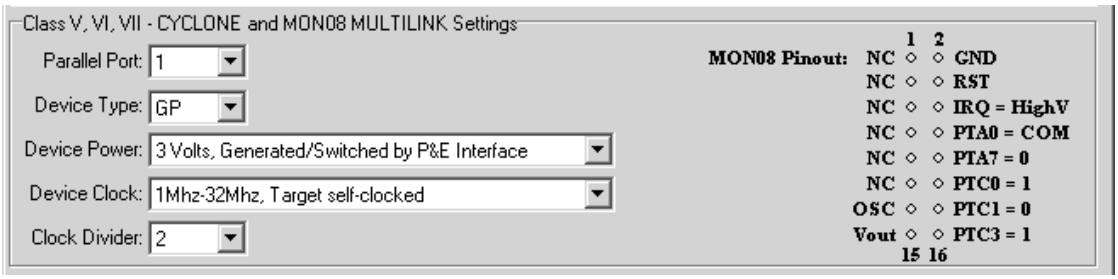

Figure 2-4: MON08 MULTILINK Pinout Example

2.6 Parallel Port

The MON08 MULTILINK connects to the PC via the 25-pin female parallel port connector. The BIOS settings for the parallel port should be one of the

following: SPP, Normal, Standard, Output Only, Unidirectional, AT. Try to avoid ECP, EPP, or PS/2 bi-directional.

If a parallel port extension cable is used, the parallel cable connecting the PC parallel port and the MON08 MULTILINK cable must be IEEE1284 compliant.

3 TARGET MON08 HEADER PINOUTS

This chapter details the MON08 connector signals according to the individual target MCU types.

3.1 68HC908AB

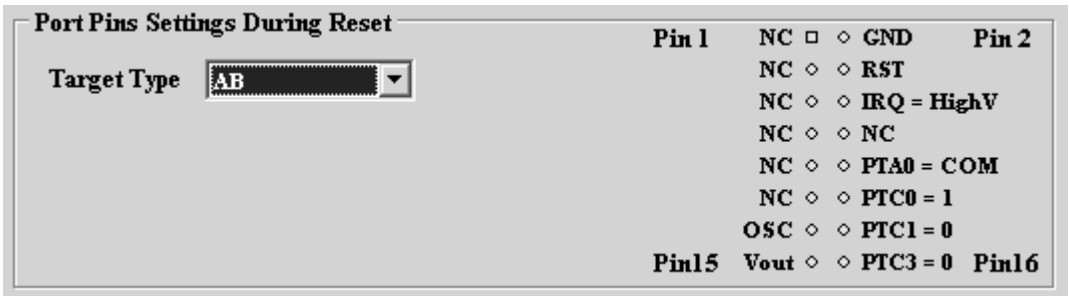

Figure 3-1: 68HC908AB Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0 and pull down PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.2 68HC908AP

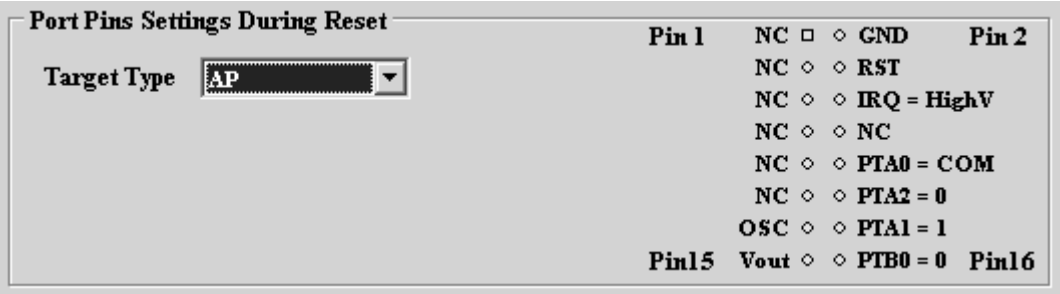

Figure 3-2: 68HC908AP Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA2, PORTA1 and PORTB0 are used for entering monitor mode. By default the user may bring these signals out to the target MON08 Header.

Alternatively, the user may pull down PORTA2 and pull up PORTA1, and pull up/down PORTB0 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.3 68HC908AS

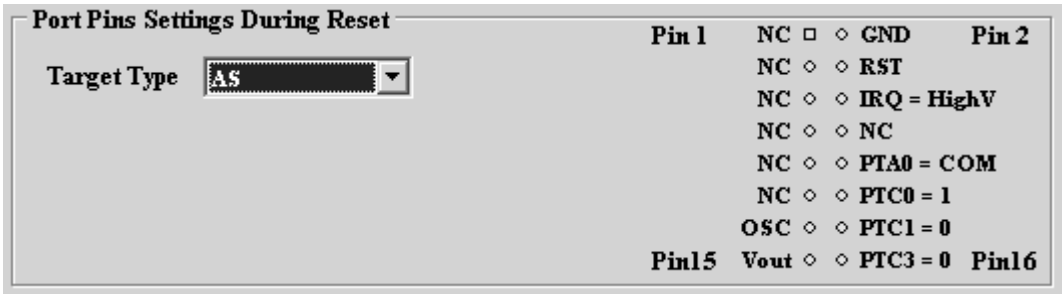

Figure 3-3: 68HC908AS Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0 and pull down PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.4 68HC908AT

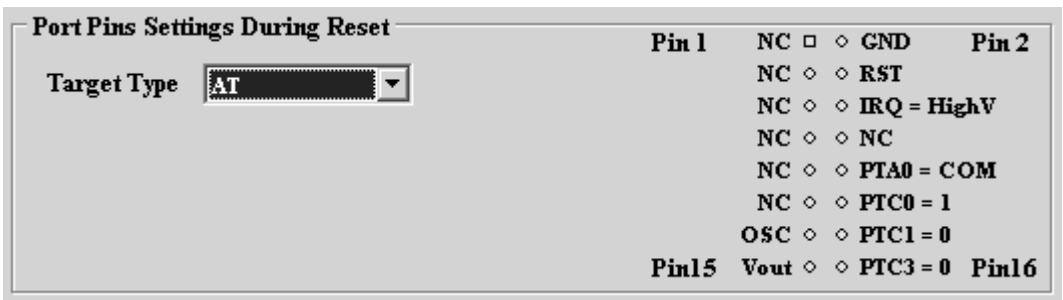

Figure 3-4: 68HC908AT Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0 and pull down PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.5 68HC908AZ

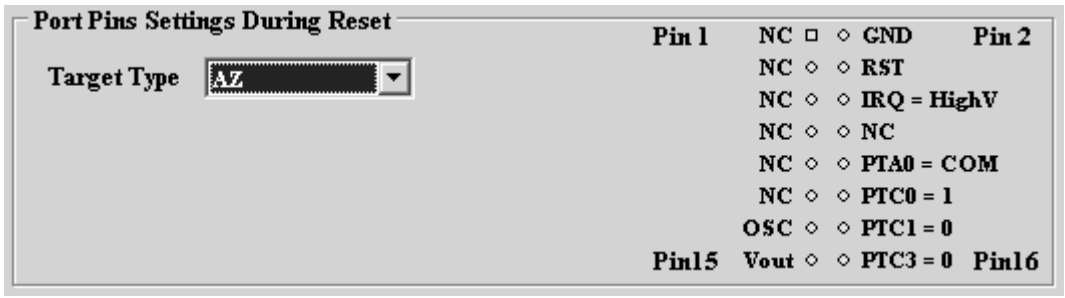

Figure 3-5: 68HC908AZ Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0 and pull down PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.6 68HC908BD

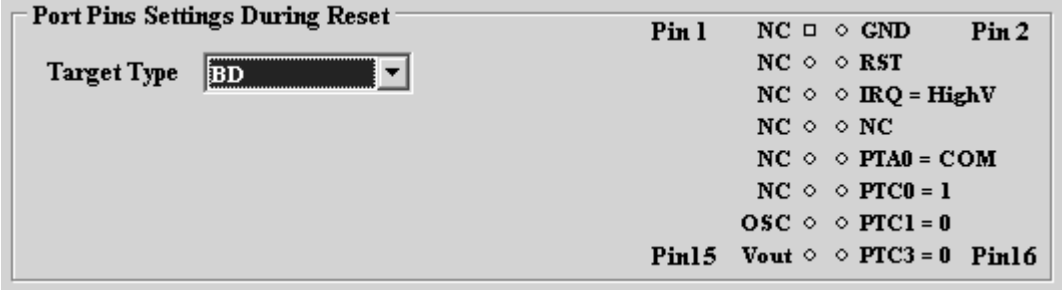

Figure 3-6: 68HC908BD Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0 and pull down PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.7 68HC908EY

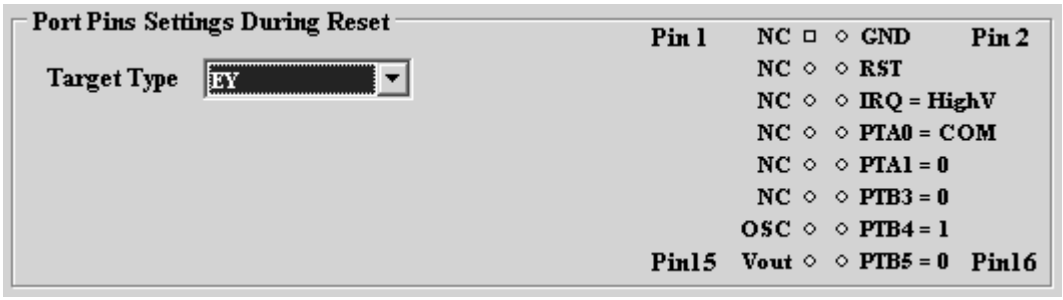

Figure 3-7: 68HC908EY Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTB3, PORTB4 and PORTB5 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull down PORTA1 and PORTB3, and pull up PORTB4, and pull up/down PORTB5 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.8 68HC908GP

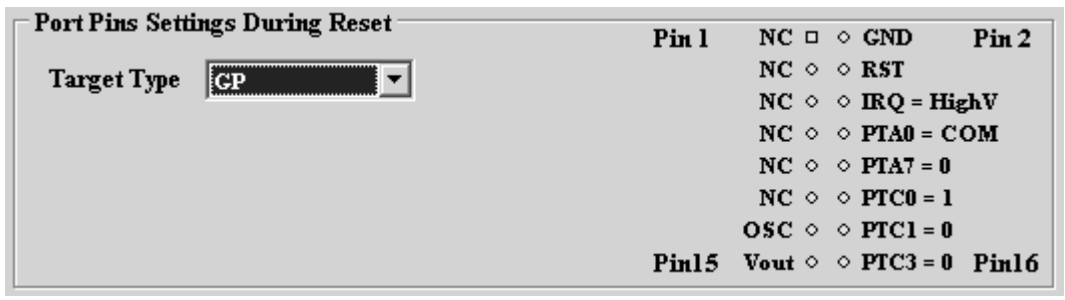

Figure 3-8: 68HC908GP Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA7, PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull down PORTA7 and PORTC1, pull up PORTC0, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.9 68HC908GR16

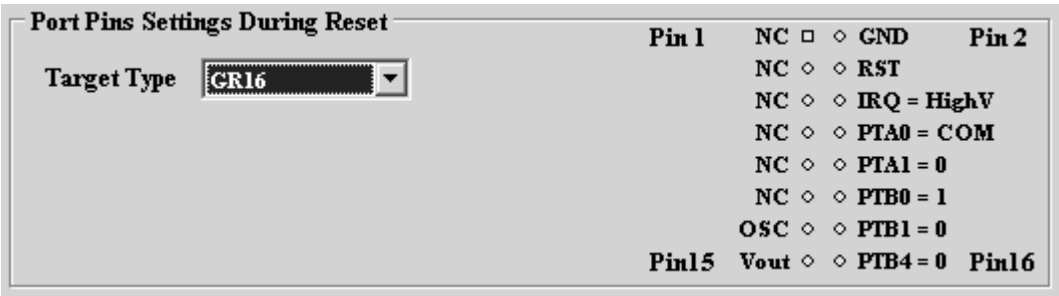

Figure 3-9: 68HC908GR16 MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTB0, PORTB1 and PORTB4 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB0, pull down PORTA1 and PORTB1, and pull up/down PORTB4 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.10 68HC908GR4/8

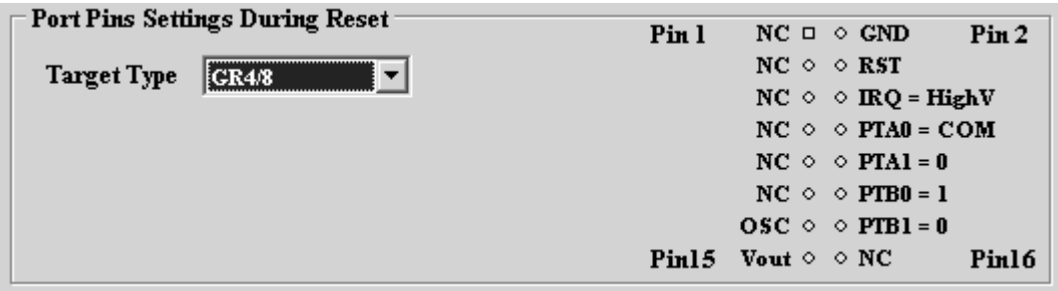

Figure 3-10: 68HC908GR4/8 MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTB0, and PORTB1 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB0, pull down PORTA1 and PORTB1. In which case the user does not need to connect these signals to the target MON08 Header. The clock division is fixed Div 4.

3.11 68HC908GT

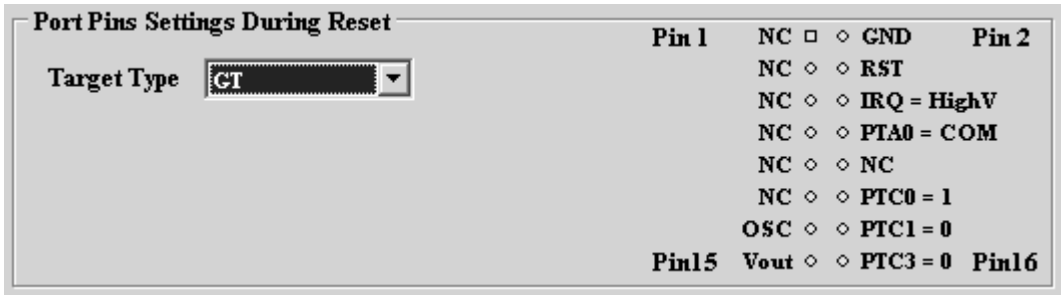

Figure 3-11: 68HC908GT Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0 and pull down PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.12 68HC908GZ

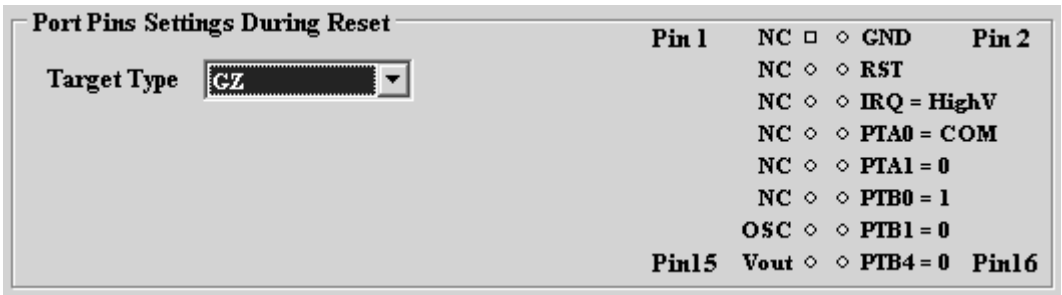

Figure 3-12: 68HC908GZ Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTB0, PORTB1 and PORTB4 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB0, pull down PORTA1 and PORTB1, and pull up/down PORTB4 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.13 68HC908JB1/8

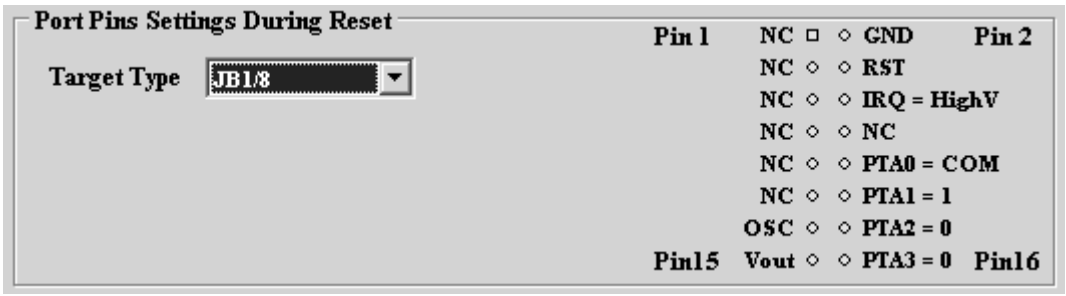

Figure 3-13: 68HC908JB1/8 MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTA2 and PORTA3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and pull down PORTA2, and pull up/down PORTA3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.14 68HC908JB16

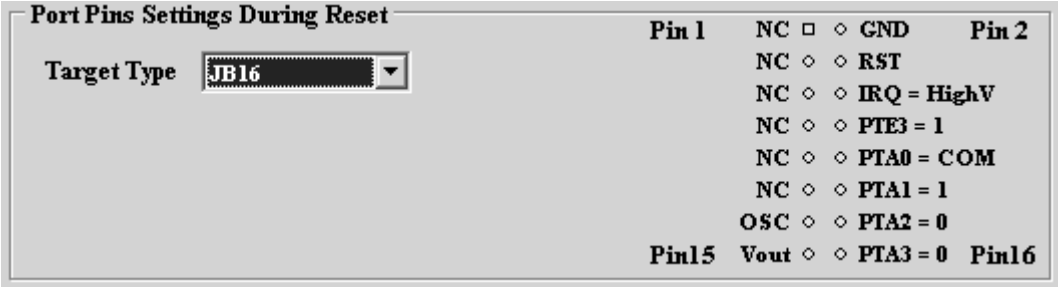

Figure 3-14: 68HC908JB16 MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTA2, PORTA3 and PORTE3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and PORTE3, pull down PORTA2, and pull up/down PORTA3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.15 68HC908JG

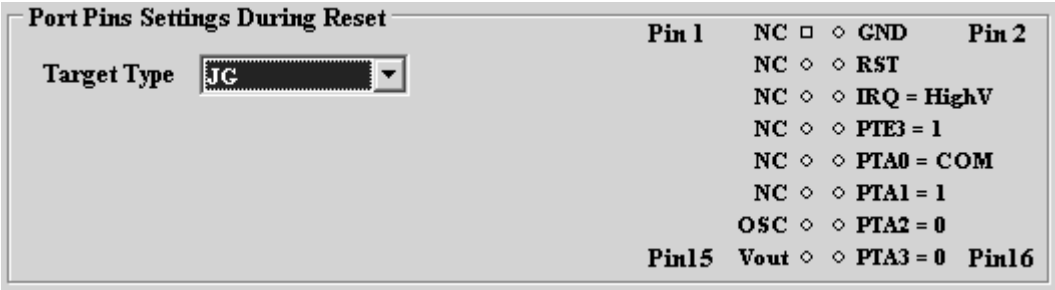

Figure 3-15: 68HC908JG Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTA2, PORTA3 and PORTE3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and PORTE3, pull down PORTA2, and pull up/down PORTA3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.16 68HC908JK

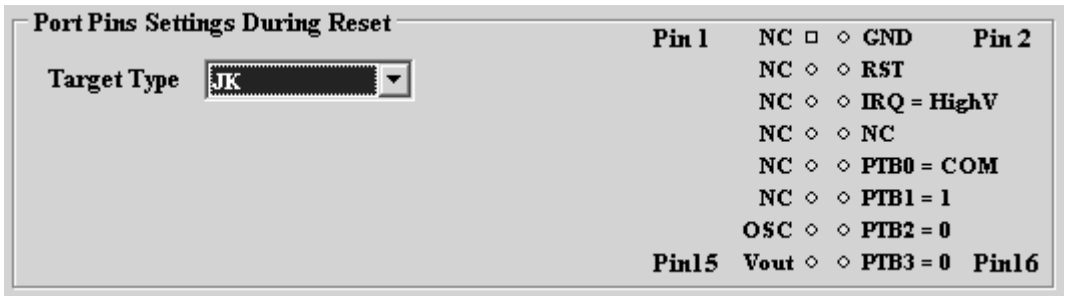

Figure 3-16: 68HC908JK Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTB0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTB1, PORTB2 and PORTB3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB1 and pull down PORTB2, and pull up/down PORTB3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.17 68HC908JL

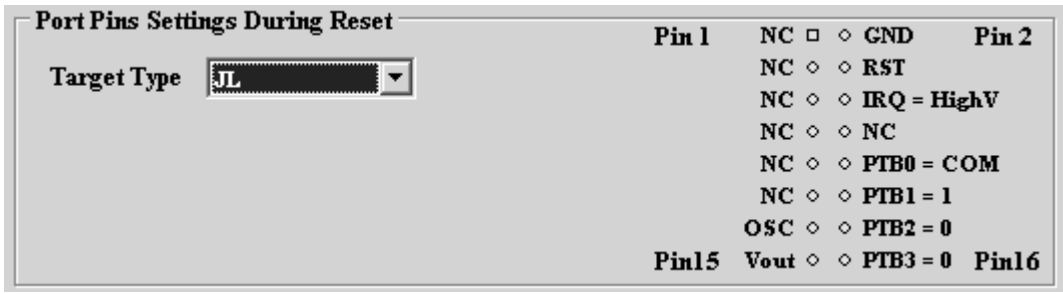

Figure 3-17: 68HC908JL Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTB0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTB1, PORTB2 and PORTB3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB1 and pull down PORTB2, and pull up/down PORTB3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.18 68HC908KX

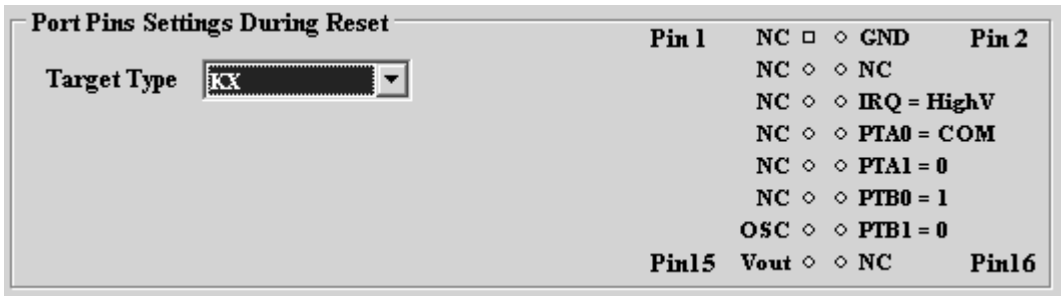

Figure 3-18: 68HC908KX Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The user must pull up the RESET line to target VDD with an external resistor.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTB0 and PORTB1 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB0 and pull down PORTA1 and PORTB1. In which case the user does not need to connect these signals to the target MON08 Header. The clock division is fixed Div 4.

3.19 68HC908LD

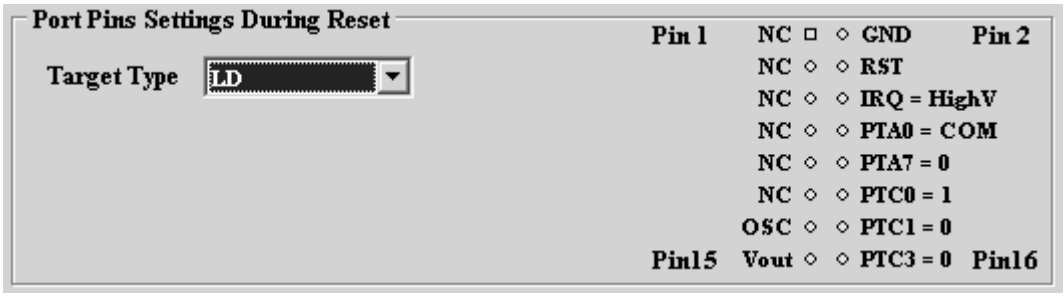

Figure 3-19: 68HC908LD Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA7, PORTC0, PORTC1 and PORTC3 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC0, pull down PORTA7 and PORTC1, and pull up/down PORTC3 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.20 68HC908LJ

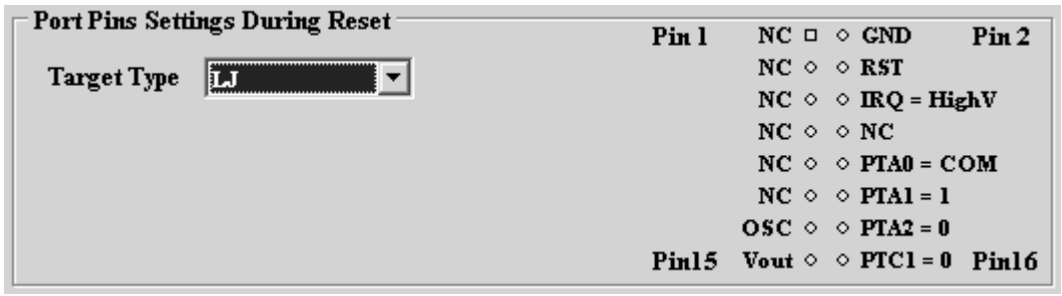

Figure 3-20: 68HC908LJ Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTA2 and PORTC1 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and pull down PORTA2, and pull up/down PORTC1 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.21 68HC908MR4/8

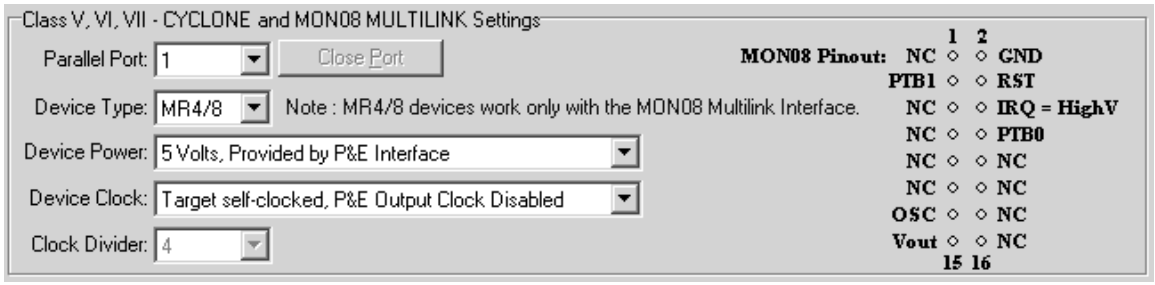

Figure 3-21: 68HC908MR4/8 Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTB0 from the target processor is connected to Pin 8 of the target MON08 Header. PORTB1 of the target processor is connected to Pin 3 of the target MON08 header. Together they serve as the data communication lines. The user

should pull down PORTB1 with a 4.7K Ohm resistor.

Please note that the MR4/8 is not currently supported by the MON08 CYCLONE.

3.22 68HC908MR16/32

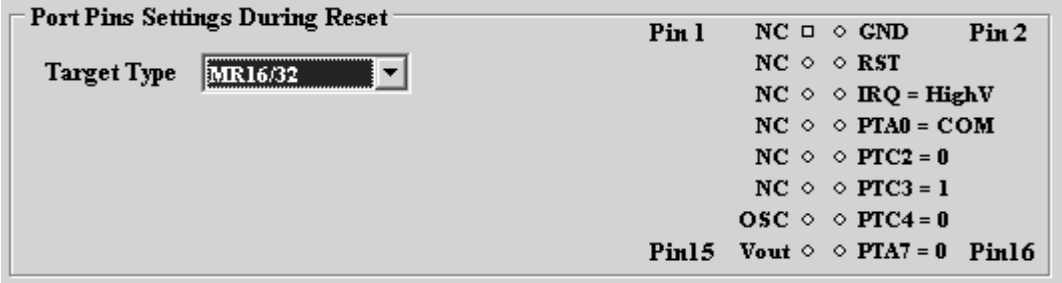

Figure 3-22: 68HC908MR16/32 Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA7, PORTC2, PORTC3 and PORTC4 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTC3, pull down PORTA7 and PORTC4, and pull up/down PORTC2 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

3.23 68HC908QT

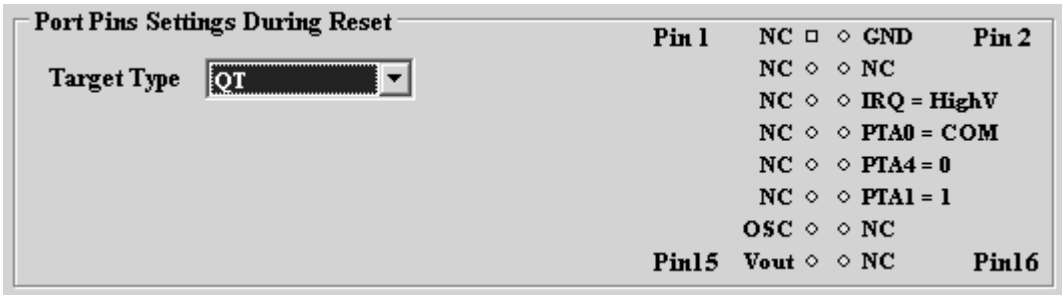

Figure 3-23: 68HC908QT Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The user must pull up the RESET line to target VDD with an external resistor.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1 and PORTA4 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and pull down PORTA4. In which case the user does not need to connect these signals to the target MON08 Header. The clock division is fixed Div 4.

Please note that the MON08 MULTILINK will calculate the proper trim value for the device being programmed and program this trim value to $FFC0.

3.24 68HC908QY

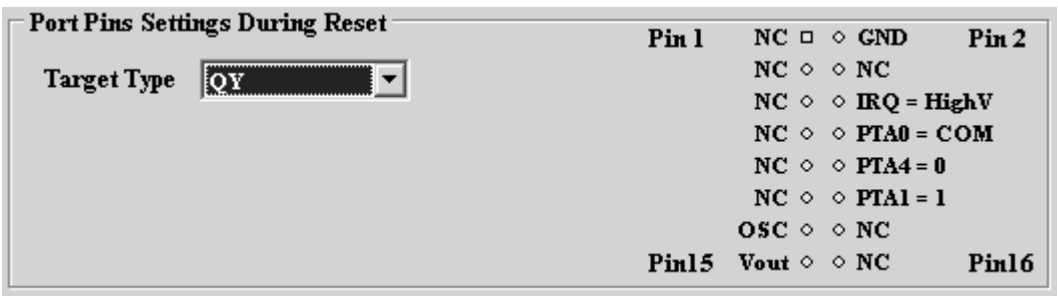

Figure 3-24: 68HC908QY Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The user must pull up the RESET line to target VDD with an external resistor.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 8, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1 and PORTA4 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and pull down PORTA4. In which case the user does not need to connect these signals to the target MON08 Header. The clock division is fixed Div 4.

Please note that in Stand-Alone programming mode the MON08 MULTILINK will calculate the proper trim value for the device being programmed and program this trim value to $FFC0.

3.25 68HC908RF

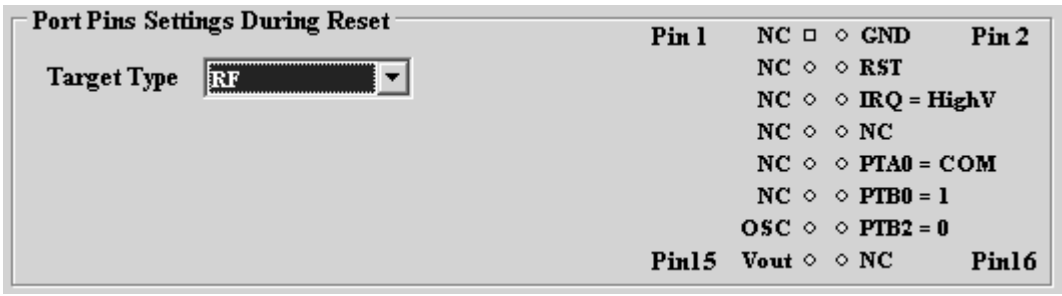

Figure 3-25: 68HC908RF Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTB0 and PORTB2 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB0 and pull down PORTB2. In which case the user does not need to connect these signals to the target MON08 Header. The clock division is fixed Div 4.

3.26 68HC908RK

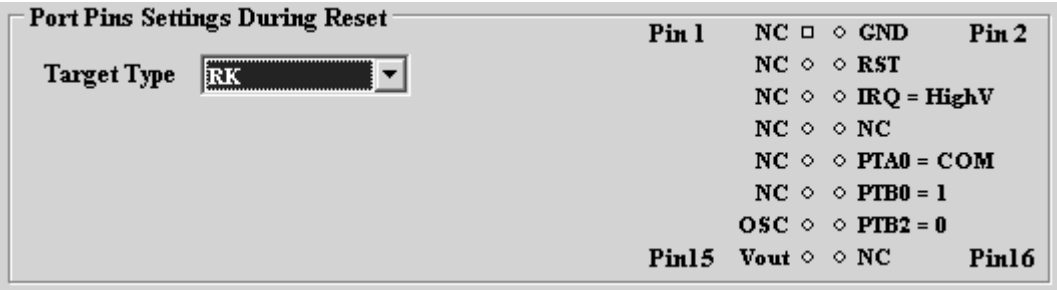

Figure 3-26: 68HC908RK Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTB0 and PORTB2 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTB0 and pull down PORTB2. In which case the user does not need to connect these signals to the target MON08 Header. The clock division is fixed Div 4.

3.27 68HC908SR

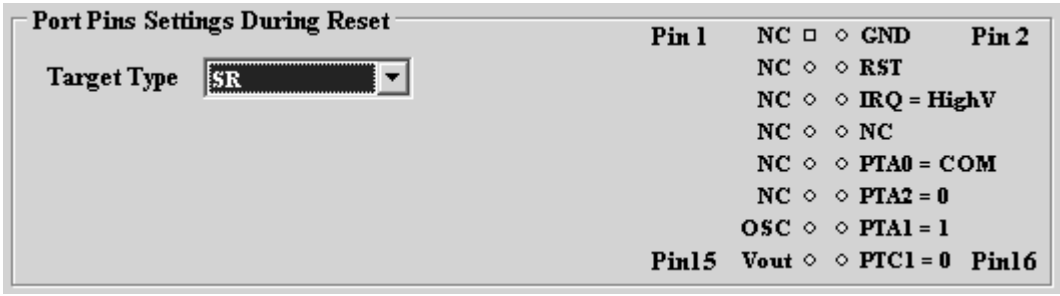

Figure 3-27: 68HC908SR Family MON08 Pinout

The target GND is connected to the Pin 2 of the target MON08 Header.

The target RESET line is directly connected to the Pin 4 of the target MON08 Header.

Alternatively, the user may pull up the RESET line to target VDD. In which case the user does not need to connect this signal to the target MON08 Header.

The target IRQ line is directly connected to the Pin 6 of the target MON08 Header.

PORTA0 from the target processor is connected to the target MON08 Header Pin 10, acting as the communications line. The MON08 MULTILINK pulls up this signal with a 10K Ohm resistor to the target VDD.

PORTA1, PORTA2 and PORTC1 are used for entering monitor mode. By default the user may directly bring these signals out to the target MON08 Header.

Alternatively, the user may pull up PORTA1 and pull down PORTA2, and pull up/down PORTC1 for clock division. In which case the user does not need to connect these signals to the target MON08 Header.

4 PC-HOSTED DEBUG/PROGRAMMING SOFTWARE

Free or low-cost software options for interactively programming and debugging 68HC08 MCUs from the PC are available from P&E Microcomputer Systems (www.pemicro.com) and Metrowerks (www.metrowerks.com). P&E's ICS08 interface software packages are available at no charge from their web site. Metrowerks' CodeWarrior Development Studio for 68HC08, Special Edition, is available at no charge from the Motorola MCU Web site (www.motorola.com/semiconductors/mcu). You must register for the license key for this software.

Note: The user should make sure they have the most recent version of these software kits. The latest updates can be downloaded from the web pages listed in **Section 4.1.1 Latest Updates - P&E Software** and **Section 4.2.1 Latest Updates - Metrowerks Software**.

4.1 P&E Microcomputer Systems Software

P&E's ICS08 software packages contain the WinIDE integrated development environment, which pulls together an assembler, in-circuit simulator, flash memory programmer, and in-circuit debugger. The programmer and debugger work with any MON08 hardware interface, including P&E's dedicated hardware interfaces. The MON08 MULTILINK is a Class V device, and the MON08 MULTILINK is Class VII.

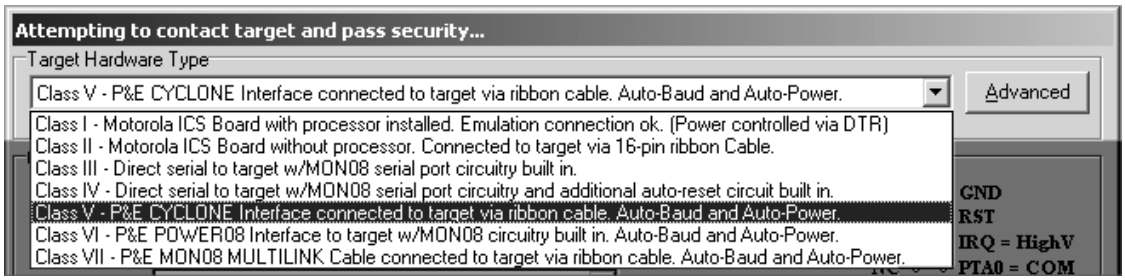

Figure 4-28: Hardware Selection in P&E PROG08SZ

4.1.1 Latest Updates - P&E Software

The most recent updates of P&E's 68HC08 software products are available to download, after a brief registration, at http://www.pemicro.com/ics08.

4.1.2 In-Circuit Debugger

The ICD08SZ In-Circuit Debugger uses the PC's Parallel Port to communicate with the MON08 MULTILINK, which further controls the target 68HC08 device via the MON08 connection. With the ICD08SZ In-Circuit Debugger you can load code into the on-chip RAM, run code out of RAM or FLASH

(already programmed by the In-Circuit Programmer), and set many software breakpoints and a single hardware (meaning in FLASH) breakpoint. The main advantage of using the ICD08SZ is that your application runs in real-time at the full bus speed of the processor.

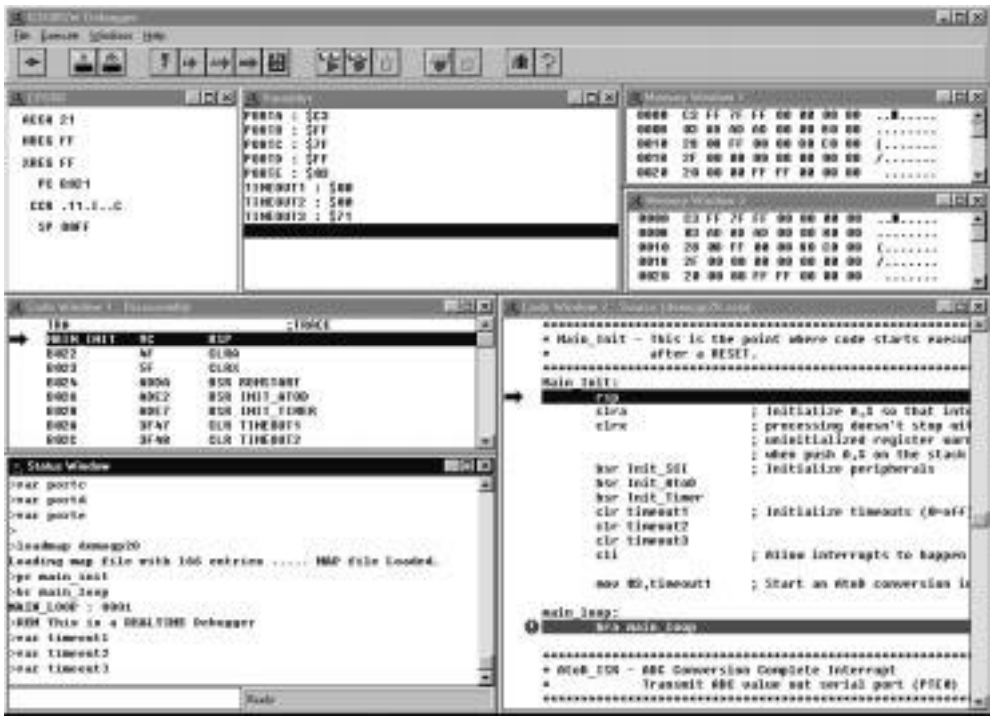

Figure 4-29: ICS08SZ Debugger Screen Snapshot

Debugger features include:

- Full-speed in-circuit emulation
- Breakpoints with counters on the Nth execution
- Variables window showing multiple data types
- Real-time execution as well as multiple tracing modes
- Startup and Macro files for automating the debug process
- Context-sensitive help for all commands
- Support for symbolic register files
- Full source-level debugging

When connecting to the target, the user will be prompted to make selections from the Target Connection And Security dialog. For more information, please see **Section 4.3 Target Connection And Security Dialog**.

4.1.3 In-Circuit Programmer

The PROG08SZ In-Circuit Programmer is a general-purpose programmer which allows the user to program any 68HC908 device with on-chip EEPROM/FLASH, either from an object file (Motorola .S19 format), or byte by byte.

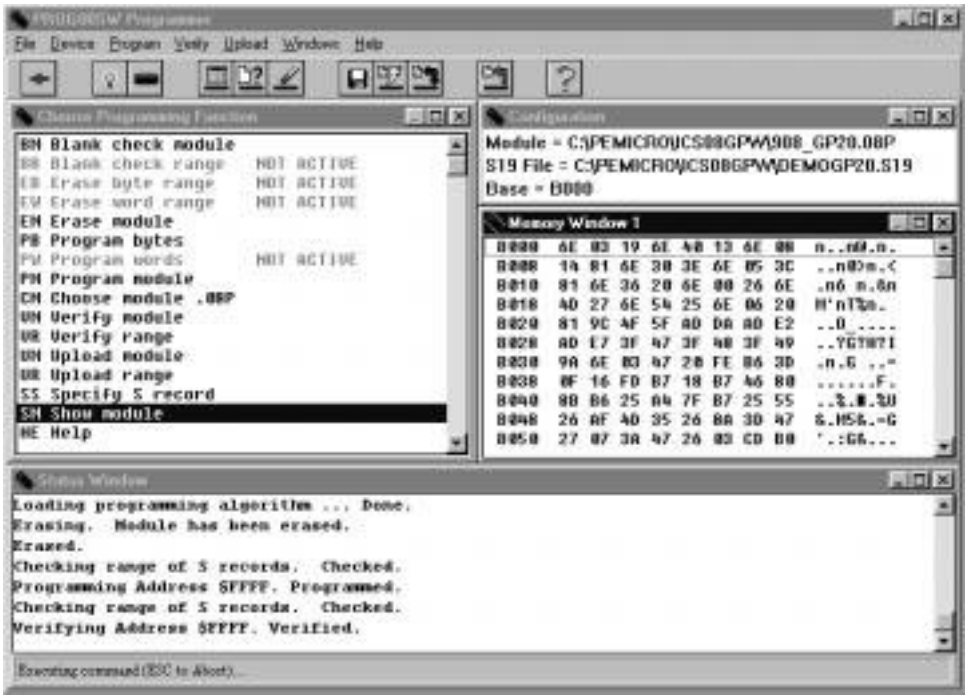

Figure 4-30: PROG08SZ Programmer Screen Snapshot

When connecting to the target, the user will be prompted to make selections from the Target Connection And Security dialog. For more information, please see **Section 4.3 Target Connection And Security Dialog**.

The PROG08SZ is simple to operate: after clicking the “Contact target with these settings” button, if the programmer successfully contacts the target it will ask you for the algorithm you wish to use during programming. Select the proper algorithm for the device you are attempting to program. Then simply select the s-record object you wish to program using the “SS” command. Now the setup of the PROG08SZ is complete and you are ready for operations on the target EEPROM/FLASH. You may choose “EM – Erase Module” to erase the target EEPROM/FLASH. Then use “BM – Blank Check Module” to see if the target EEPROM/FLASH is indeed erased. After that, you may choose “PM – Program Module” to program the S-record object into the target. Finally, you may use “VC – Verify CRC Checksum” to verify that the contents are properly programmed in the target memory.

4.1.4 Command Line Programmer

CPROG08SZ is a command line programmer that allows quick turn-around time for programming target MCUs. The user may create a script file to instruct the software to execute specific commands in sequence. Please refer to CPROG08SZ.pdf for more information.

4.2 Metrowerks Software

The special edition of Metrowerks' CodeWarrior studio offers absolute assembly and provides debugging capabilities based on P&E's programming and debug technologies.

4.2.1 Latest Updates - Metrowerks Software

The most recent updates of Metrowerks CodeWarrior software is available at: http://www.metrowerks.com/MW/Support/Download/default.htm?did=find&vers=CWHC08&submit=Find.

4.2.2 Metrowerks CodeWarrior

A programming or debug session with the project-based CodeWarrior IDE may be launched by double-clicking on the project name (format is projectname.mcp) from your file storage. Starting a new project is a little more challenging, but the tutorials, FAQs, and Quick Start Guides are easy to follow and have you building a new project, using pre-built templates, in a short time. (See www.Metrowerks.com/MW/Develop/ and select "CodeWarrior Development Studio for HC08 for Microcontrollers".)

The following example illustrates how to program and debug an M68HC908 MCU from within the CodeWarrior IDE.

Here are the main steps in programming the FLASH with CodeWarrior and starting a debug session.

1. a. Launch the CodeWarrior CW08 software and create a new project,
   or
   b. Double-click on your project file (projectname.mcp)

The Project Manager window appears. See **Figure 4-31**.

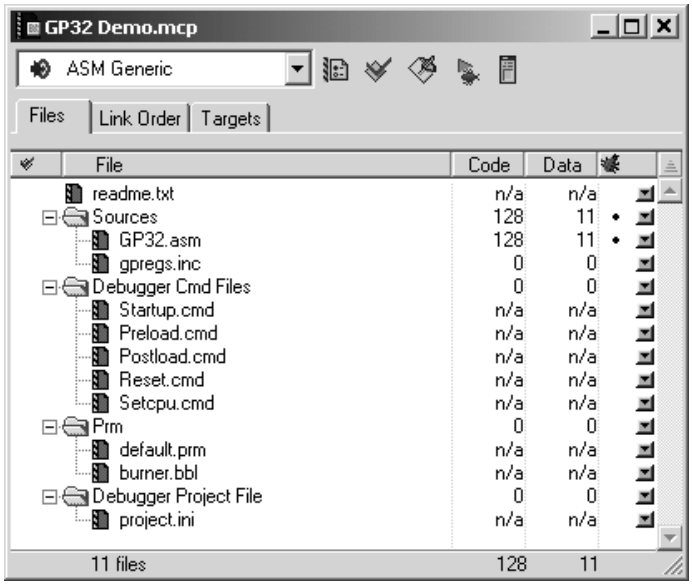

Figure 4-31: CodeWarrior Project Window

2. Click the + sign to expand the Sources folder.
3. Modify the source file if necessary.
4. Click the Debug icon (green arrow). The True-Time Simulator and Real-Time Debugger launches. See **Figure 4-32**.

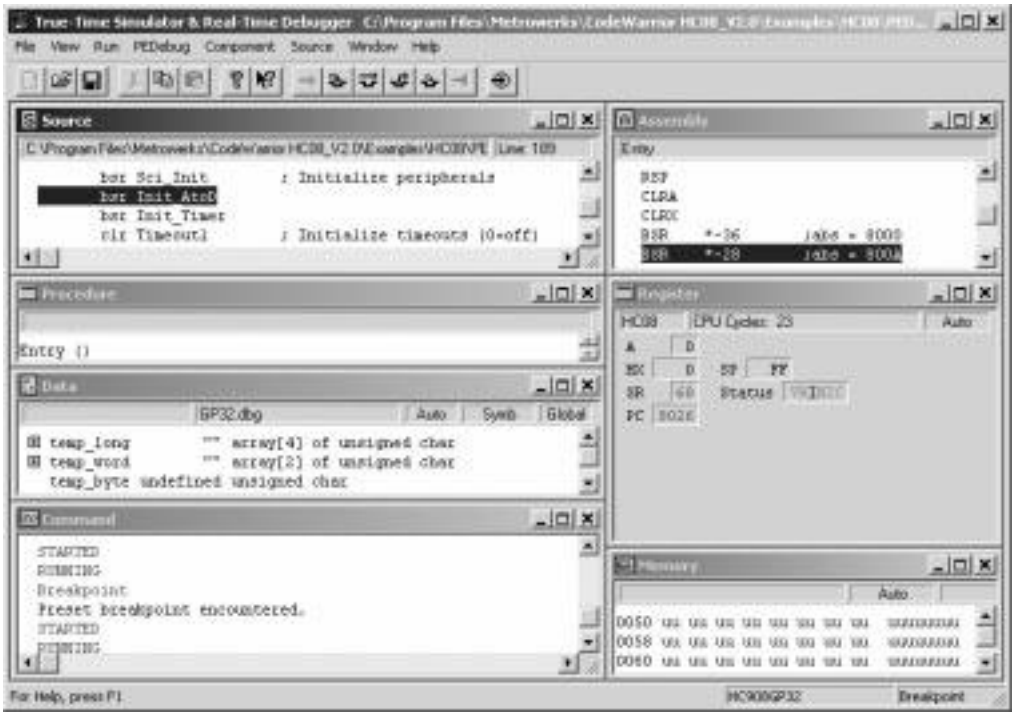

Figure 4-32: True-Time Simulator & Real-Time Debugger Window

5. Select the PEDebug pull-down menu and navigate to the appropriate device as shown in **Figure 4-33**.
6. Likewise, in the PEDebug pull-down menu, select Mode: In-Circuit Debug/Programming.

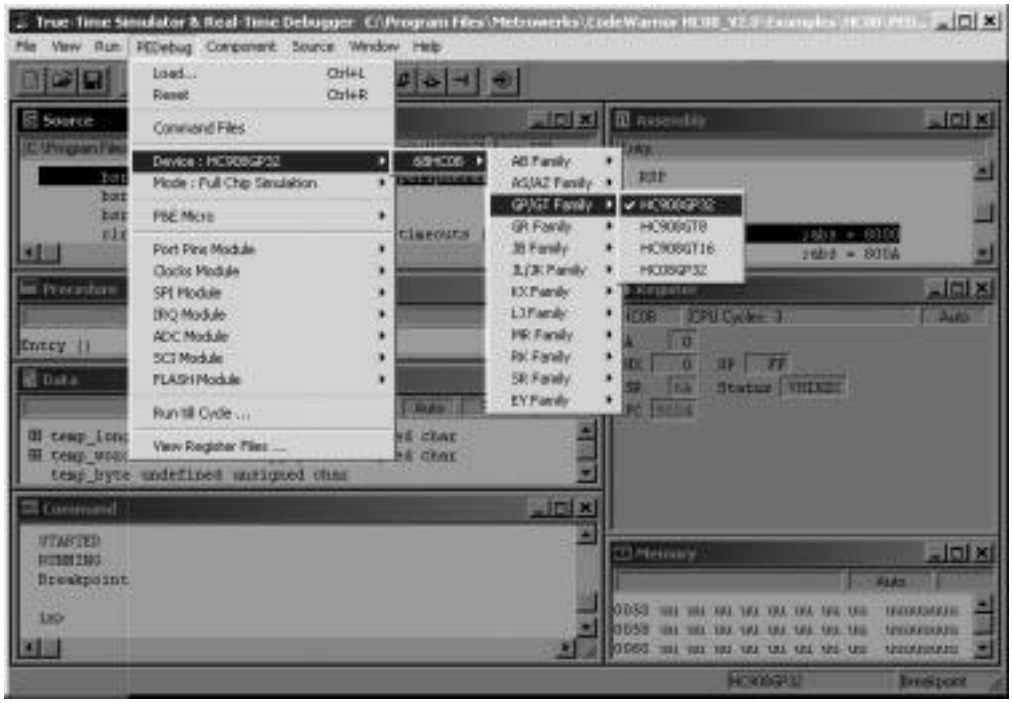

Figure 4-33: PEDebug Pull-down Menu

The PROG08SZ Attempting to contact target and pass security… window appears.

7. Select the appropriate class in Target Hardware Type (Class VII for MON08 MULTILINK and Class V for MON08 CYCLONE).
8. Click Contact target with these settings…
9. Follow the Power Cycle dialog instructions.
10. Click Yes in Confirm window. (**Figure 4-34**)
11. Click Yes in Erase and Program FLASH window. (**Figure 4-35**)
12. Follow the subsequent Power Cycle dialog instructions as the scripted procedure automatically establishes communications, erases the FLASH if necessary, and programs the FLASH.

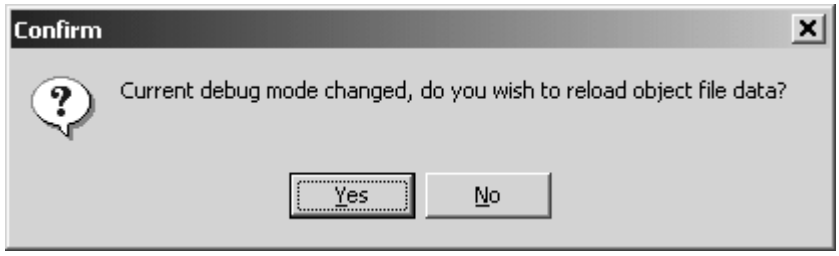

Figure 4-34: Confirm Window

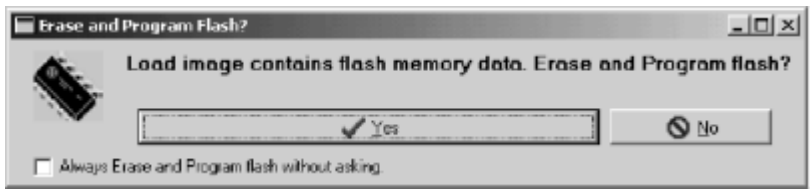

Figure 4-35: Erase And Program Flash Window

At this point, the FLASH memory is programmed and ready for debug. The True-Time Simulator & Real-Time Debugger integrates the debugger tools from P&E Microcomputer Systems in this example. The windows look slightly different between the ICD08SZ and True-Time tools but the same basic debugger (ICD08SZ) drives both.

4.3 Target Connection And Security Dialog

The following is an explanation of each part of the target connection dialog. For information on passing security mode, read this topic carefully, and refer to **Section 4.3 Target Connection And Security Dialog**.

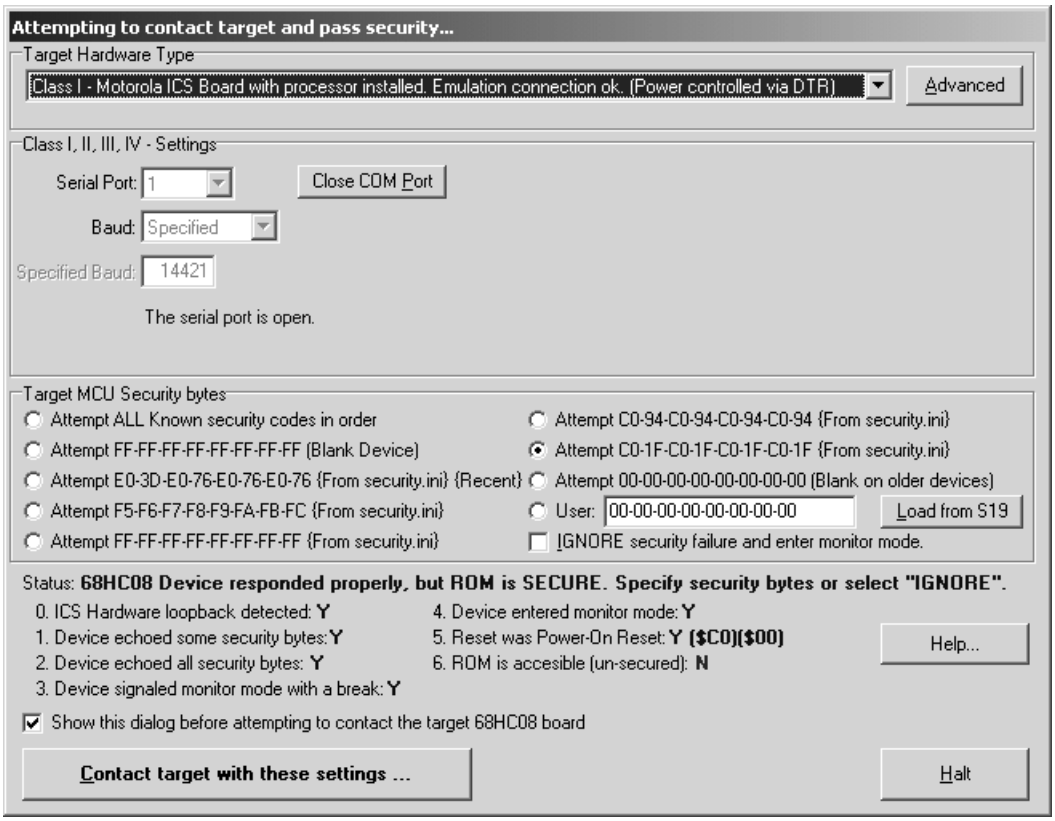

Figure 4-36: Initial Target Connection And Security Dialog Box

4.3.1 Target Hardware Type

This section of the dialog allows you to select the type of hardware configuration to which you are trying to connect, as well as modify specific protocol settings.

Note: If you select Class V, VI, or VII in the Target Hardware Type selection box, the second section of the Target Connection and Security Dialog changes. Please refer to **Figure 4-40** and **Section 4.3.1 Target Hardware Type** for a depiction and description.

4.3.1.1 Class Of Target Board

There are several different configurations of target boards, and P&E's MON08-based applications communicate to each type of hardware a little differently. The options are:

Class I

ICS Board with processor installed. This is the standard and most common configuration of the ICS08 boards. In this configuration, the processor is resident in one of the sockets on the ICS board itself. The processor can be debugged and programmed in this configuration, and an emulation cable containing all the processor I/O signals can be connected to the user's target board. In this configuration, the ICS board hardware can automatically power up and down the processor in order to pass security in the simplest fashion. The user has to be sure **not** to provide power from the target, up through the emulation cable, to the processor pins themselves, when this dialog appears. This is so that the software, when attempting to establish communications, can fully power the processor down. The software running on the PC controls power to the target via the serial port DTR line. This configuration can be specified at startup in the software by using the **ICS08** command-line parameter; otherwise the software will remember the hardware configuration from session to session.

Class II

ICS Board without processor, connected to target via MON08 Cable. In this configuration, there is no processor resident in any of the sockets of the ICS board itself. The processor is mounted down in the target system. The connection from the ICS board to the target is accomplished via the 16-pin MON08 connector. In this configuration, since the ICS does not control power to the processor, the user will be prompted to turn the processor's power supply on and off. Turning off the power supply is necessary in order to be able to pass the initial security mode check and access the flash on the processor. A simple reset is not enough; to pass the security check, you must first force the processor to encounter a POR (power-on reset) which requires that the processor's voltage dip below 0.1v. Once security has been passed, resetting the device or re-entering the software should be easier. This configuration can be specified at startup in the software by using the **MON08** command-line parameter; otherwise the software will remember the hardware configuration from session to session.

Class III

Custom Board (no ICS) with MON08 serial port circuitry built in. In this configuration, the ICS board is not used at all. The user must provide a serial port connection from the PC, and provide all hardware configuration necessary to force the processor into MON08 mode upon reset. This includes resets both internal and external to the processor. In this configuration, because the software does not directly control power to the processor, the user will be prompted to turn the processor's

power supply on and off. The use will also be prompted to turn power on and off to reset the target processor, as the PC doesn't have control of the target reset. Turning off the power supply is necessary mainly to be able to pass the initial security mode check and access the flash on the processor. A simple reset is not enough; to pass the security check, you must first force the processor to encounter a POR (power-on reset) which requires that the processor's voltage dip below 0.1v. Once security has been passed, resetting the device or re-entering the software should be easier. This configuration can be specified at startup in the software by using the **NODTR** command-line parameter; otherwise the software will remember the hardware configuration from session to session. The Class III selection also applies to use of the ICS board with the two-pin blank part programming connector.

Class IV

Custom Board (no ICS) with MON08 serial port circuitry and additional auto-reset circuit built in. In this configuration, the ICS board is not used at all. The user must provide a serial port connection from the PC and all hardware configuration necessary to force the processor into MON08 mode upon reset. In addition, the user must include an extra circuit which allows the reset line of the processor to be driven low from the DTR line of the serial port connector (Pin 4 on a DB9). The following diagram shows the additional connection needed to reset from a DB9 serial connector.

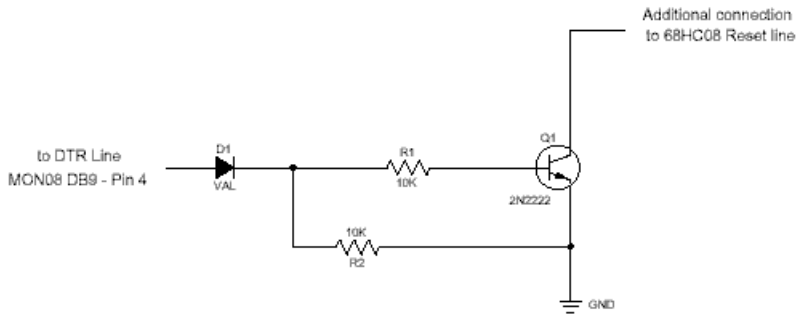

Figure 4-37: Additional Connection To Reset From DB9

In this configuration, because the software does not directly control power to the processor, the user will be prompted to turn the processor's power supply on and off. Turning off the power supply is necessary in order to be able to pass the initial security mode check and access the flash on the processor. A simple reset is not enough; to pass the security check, you must first force the processor to encounter a POR (power-on reset) which requires the processor's voltage to dip below 0.1v. Once security has been passed, resetting the device

should be facilitated by the above circuitry. This configuration can be specified at startup in the software by using the **NODTRADD** command-line parameter; otherwise the software remembers the hardware configuration from session to session.

Class V

P&E MON08 CYCLONE connect to target via ribbon cable. Allows Auto-Baud and Auto-Power.

Figure 4-38: MON08 Cyclone MON08 Interface and Stand-Alone Programmer

P&E's MON08 Cyclone is a stand-alone automated programmer and MON08 interface. This unit can be used as a debug and programming interface with P&E software applications on the PC, or it can be pre-programmed and used in stand-alone mode.

Class VII

P&E MON08 Multilink Cable connect to target via ribbon cable. Allows Auto-Baud and Auto-Power.

Figure 4-39: MON08 Multilink Interface Cable

The MON08 MULTILINK is an interface cable whose purpose is to allow debug and programming of 68HC08 devices via the MON08 debug port. The

MON08 MULTILINK connects the target to the PC via a standard parallel port.

Note: If you select Class V, VI, or VII in the Target Hardware Type selection box, the second section of the Target Connection and Security Dialog changes. Please refer to **Figure 4-40** and **Section 4.3.1 Target Hardware Type** for a depiction and description.

Also:

For the simulator, the **/SIM08** command-line parameter causes the software to disconnect from the target and enter Simulation Only mode.

For information on passing security mode, read this topic carefully and also refer to **Section 4.3.5 68HC08 SECURITY MODE**.

4.3.1.2 Class V, VI, VII Options

If you select Class V, VI, or VII in the Target Hardware Type selection box, the second section of the Target Connection and Security Dialog changes to appear as below.

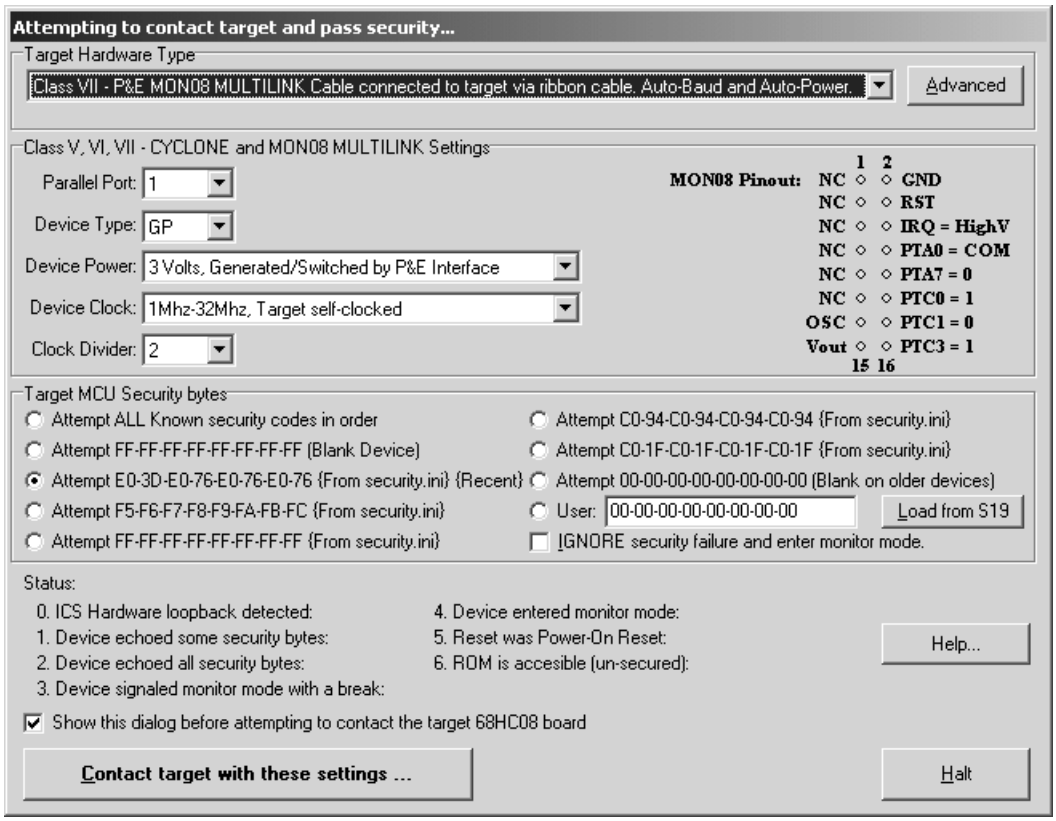

Figure 4-40: Class V, VI, VII Target And Security Dialog

The options presented to the user are as follows:

Device Type

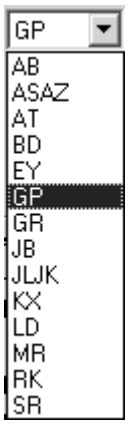

Figure 4-41: Device Type Selection Box

The device type selection box allows the user to specify what type of HC08 they are communicating with. The dialog will then display the appropriate pinout to be implemented on the MON08 connector, so that the P&E interface can talk to it properly. The values given (1 or 0) are for informational purposes only and are driven by the P&E interface.

Device Power

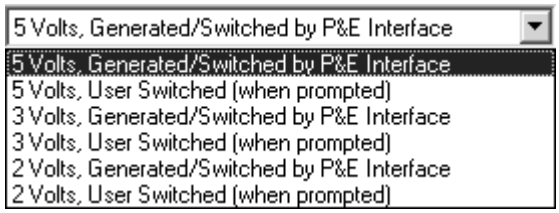

Figure 4-42: Device Power Dialog

The device power selection allows the user to specify whether the target is 2, 3, or 5 Volts, and whether this power is switched/generated by the P&E interface or if it is separately supplied to the target and under user control. If it is under user control, the software will use dialog boxes to ask the user to power the target up and down when necessary (similar to Class II-IV).

Device Clock

Figure 4-43: Device Clock Selection Box

The device clock menu allows three options:

1) P&E provides clock to target

2) The target has its own clock (1-32MHz)

3) The target has a slow crystal (30KHz-100KHz) with PLL circuitry. P&E tries to enable the PLL to allow programming and debug at higher speeds.

Baud

There is no need to set baud rate for Class V, VI, or VII targets, as it is auto-detected from the target.

4.3.1.3 Advanced Settings Dialog

The Advanced Button brings up a dialog which allows the user to set specific protocol settings. The following is an explanation of each part of the advanced

settings dialog.

Figure 4-44: Target Hardware Type: Advanced Settings Dialog

Tpd and Tpu Timing

These timing parameters are mostly designed for Class I boards, although the delays are valid for all classes of boards. Many of the ICS boards and user target boards need time to power down and power up.

Whenever power is automatically switched off, or is manually requested to be switched off, the software waits for an amount of time equal to the Tpd delay time before proceeding to the connection protocol. This is because a board or power supply may have capacitance which holds the power up for a short time after the supply has been switched off, but the supply voltage must reach less than 0.1v before it is turned back on if a Power-On reset is to occur.

Whenever power is automatically switched on, or is manually requested to be switched on, the software waits for an amount of time equal to the Tpu delay time before attempting to contact the 68HC08 processor. This is to allow time not only for power to be fully available, but to wait until any reset driver has finally released the RESET line. On many ICS08 boards (such as the ICS08RK, M68ICS08JL3, M68ICS08JLJK, and ICS08GP20) the Tpu can be decreased to as little as 250ms with no adverse effects.

Target has RESET button (class III boards only): The software occasionally needs to get control of the target. On systems which are Class III boards with the monitor mode circuitry built-in (including RS-232 driver), there is no means to reset the target to gain control. If the board has a reset button, the software can use this to gain control of the target system. If this option is checked, the software will prompt the user to push the target reset button when a reset of the target system is desired. If the option is unchecked, the software will ask the user to power cycle the target system to achieve a

reset.

MON08 Cable connection communications type (Class II boards Only)

This selection box is valid only for Class II hardware configurations using the MON08 cable. It allows the user to specify the sequence that the software uses to power up the ICS system. When the software tries to create a power-on reset condition, two events must occur:

1. Power of the target MCU must go below 0.1v. This means that the processor can not be receiving power from its power pins, nor can it have a significant voltage being driven on port pins or the IRQ line, as these will drive the MCU power back through these pins. It is crucial, therefore, to have the ICS and the Target both powered down at some point in time.

2. The processor MON08 configuration pins, including IRQ, must be properly driven when the target processor resets to drive it into monitor mode. If these pins are not set up properly before the processor powers up, the processor may start up in user mode.

Power Down ICS, Ask the user to power down their board, Power Up ICS, Ask the user to power up their board

This is the default option and should work for most, if not all, ICS08/Target Board solutions. Refer to the manual addendum under startup for the settings for a specific ICS board. It requires the user go through two dialog stages, and requires more time than simply cycling the power.

1. Software automatically powers down the ICS.

2. Software Asks the user to power down the board as follows:

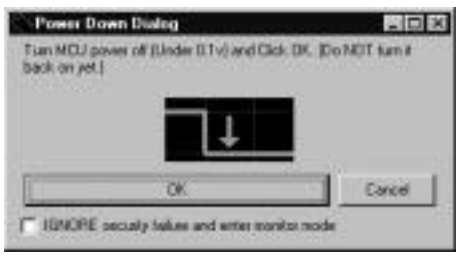

Figure 4-45: Power Down Dialog

3. Software automatically powers up the ICS, which configures the processor's MON08 configuration pins.

4. Software asks the user to power up the board as follows:

Figure 4-46: Power Up Dialog

Power Down ICS, Ask the user to power cycle their board, Power UP ICS

This option will work for many ICS boards as well, but relies on the fact that while the ICS is powered off, it will hold the target in reset until it is powered up itself and has configured the MON08 configuration pins. The sequence of events in this mode is:

1. Software automatically powers down the ICS.
2. Software asks the user to power cycle their board as follows:

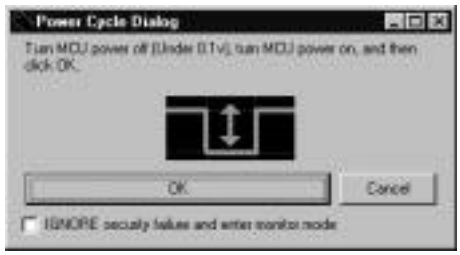

Figure 4-47: Power Cycle Dialog

3. Software automatically powers up the ICS, which configures the processors MON08 configuration pins.

4.3.2 TARGET MCU SECURITY BYTES

One of the steps that is necessary to properly bypass security is to provide the proper security code for the information that is programmed into the part. This holds true even when the part is blank.

The security code consists of the 8 values which are currently stored in flash locations $FFF6 - $FFFD of the processor. The PROG08SZ flash programming software continually records any changes to these security bytes and stores them in the file SECURITY.INI. The information in this file is shared with P&E's In-Circuit Debugger and In-Circuit Simulator software, and will appear in the dialog box. This allows the user to specify which security code to use to pass security.

This dialog can also be used by the user to manually enter the proper security bytes via the USER setting, or to load the security bytes from the same .S19 file which was programmed. The bytes are loaded from an .S19 file by clicking the “Load from S19” button.

IGNORE security failure and enter monitor mode

This checkbox can be used to cause the software to ignore a failure to properly pass the 68HC08 security check. If the checkbox is set, the software will attempt to establish monitor mode communications regardless of the security status. As long as the Baud and Port are correct, and the device has been properly powered, this will allow monitor mode entry. Note that by ignoring the security check failure, you may use monitor mode, but the ROM/Flash will not be accessible.

The checkbox can be set to be checked on startup via the **FORCEBYPASS** command-line parameter, which will cause the software to ignore security check failure. This checkbox can be overridden to be *un*checked on startup via the **FORCEPASS** command-line parameter, which will cause the software to pop-up the connection dialog when the security check has failed. Note that if a connection is not established for a reason other than security failure, the connection dialog will always appear.

4.3.3 STATUS

The status area consists of one status string following the “Status:” label, and seven items which list the state of the last attempt to connect to a target and pass security. The description for these items is as follows:

0 – ICS Hardware loopback detected:

Every ICS or board which supports MON08 has a serial loopback in hardware which, by connecting the transmit and receive lines, automatically echoes characters from the PC. A valid character transmitted from the PC should be echoed once by the loopback circuitry on the board and once by the monitor of the target processor itself. This status indicates whether or not the first echoed character from the hardware loopback was received when one of the security bytes was transmitted. If the status is ‘N’, which indicates that the character was not received, it is most likely due to one of the following reasons:

1. Wrong Com Port specified.
2. The baud rate specified was incorrect (probably too low).
3. The ICS/Target is not connected.
4. No Power to the ICS.

If this status bit responded with an ‘N’, you must correct this before

analyzing the reset of the status bits.

1 – Device echoed some security bytes:

The monitor resident in a 68HC08 device automatically echoes every incoming character when it is in monitor mode. A valid character transmitted from the PC should be echoed once by the loopback circuitry on the board and once by the monitor of the target processor itself. This status indicates whether or not the second echoed character from the monitor response was received when one of the security bytes was transmitted. If the status is 'N', which indicates that the character was not received, or not received properly, it is most likely due to one of the following reasons:

1. The baud rate specified was incorrect.
2. The part did not start the monitor mode security check on reset. Signals to force monitor mode may be incorrect.
3. No Power to the ICS.

If this status bit responded with an 'N', you must correct this before analyzing the reset of the status bits.

2 – Device echoed all security bytes:

In order to pass security, the software must send 8 security bytes to the processor. The processor should echo each of these eight bytes twice. If all 8 bytes did not get the proper two-byte echo, this flag will be 'N'. Reasons for this include:

1. The part did not start the monitor mode security check on reset. Signals to force monitor mode may be incorrect.
2. The baud rate specified was incorrect.
3. The processor was not reset properly. Check the "Target Hardware Type" and if you are connecting to a class II board, check the "MON08 cable communication connections type" in the "advanced settings" dialog.

3 – Device signaled monitor mode with a break:

Once the processor has properly received the 8 bytes from the PC software to complete its security check, it should transmit a break character to the PC signaling entry into monitor mode. This break should be sent regardless of whether the security check was successfully passed. If a break was not received from the processor, this flag will be 'N'. Reasons for this include:

1. The baud rate specified was incorrect.
2. The processor was not reset properly. Check the "Target Hardware

Type". If you are connecting to a class II board, check the "MON08 cable communication connections type" in the "advanced settings" dialog.

4 – Device entered monitor mode:

Once the software has received, or failed to receive, a break from the processor, it attempts to communicate with the monitor running on the 68HC08 processor. It tries to read the monitor version number by issuing a monitor mode read. If the processor fails to respond properly to this command, this flag will be 'N'.

5 – Reset was Power-On Reset:

If the device properly entered monitor mode (4), the software will read the reset status register (RSR). This read does not affect the security sequence, and occurs purely for diagnostic reasons. The reset status register indicates the conditions under which the processor underwent the last reset. For the software to pass the security check properly, it MUST first cause the processor to undergo a Power-On Reset. The software reads the reset status register to determine if the last reset was indeed caused by power-on. The result of the reset status register is indicated in parentheses after the flag value. If the highest bit is not set then the reset was not a power on reset, and the flag will indicate 'N'. Reasons for this include:

1. The processor did not power all the way down because power was being supplied to the processor through either the port pins, IRQ line, RESET line, or power pins.

2. The voltage driven on the power pin of the processor did not go below 0.1 volts.

3. The processor was not reset properly. Check the "Target Hardware Type". If you are connecting to a class II board, check the "MON08 cable communication connections type" in the "advanced settings" dialog.

6 – ROM is accessible (un-secured):

If the device properly entered monitor mode (4), the software reads locations $FFF6-$FFFF to determine if the processor passes the security check. Memory locations which are invalid or protected read back from the device as $AD. If all bytes from $FFF6-$FFFF read a value of $AD, it is assumed the device is secure, and the flag value is an 'N'. If all flags 0-5 register a value of 'Y' and flag 6 register a value of 'N,' then the reset process has gone correctly except that the security code used to pass security was incorrect. Specify the correct security code and try again, or IGNORE the security failure and erase the device. Once you erase a secured device, you must exit the software and restart it in order to pass

security.

4.3.4 ADDITIONAL DIALOG BUTTONS

The following buttons are also available:

Contact target with these settings – This causes the software to attempt to cause a power on reset of the target, and to attempt to pass security with the settings in this dialog.

Simulation Only – This button is only visible in In-Circuit Simulation. This causes the In-Circuit Simulator not to use the target and, instead, to do completely software-based simulation. The **/SIM08** command-line parameter has the same function.

Halt – This causes the software to terminate and return to the calling environment.

4.3.5 68HC08 SECURITY MODE

Monitor mode is a special mode on the 68HC08 device which allows an external host to control the 68HC08 microcontroller via an asynchronous serial interface. This feature allows a host computer to query and modify the state of the processor including to load, debug, and program code. Without any protection mechanism, this same feature could be used to read out the internals of the microcontrollers ROM.

The M68HC08 microcontrollers have a additional built-in mechanism to protect a programmed device from being read and disassembled. The mechanism allows a user who knows the security unlock code to enter monitor mode and access the internal ROM/flash. This is often desirable to allow real-time debugging of a programmed device. The ICD08SZ allows just such functionality.

The security mechanism also allows a user who doesn't know the security code to enter monitor mode, but doesn't give them access to the ROM. Upon failing the security protocol, the ROM/Flash is removed from the memory map until the next POWER-ON reset, in which case the host has to bypass security again. The advantage of this is that even though any on-chip flash is not READ accessible, it is erasable. Forgotten what you programmed into your device? The answer is simple: erase it.

A device is automatically protected in this manner. The 8 bytes from address $FFF6 to $FFFD constitute the security unlock code which can be used to pass the security check and get access to the ROM/Flash. Hence, if a user knows what has been programmed into a device, they implicitly know the security unlock code.

In order to facilitate passing the security check on a 68HC08 device, the

PROG08SZ software continually records any changes to these security bytes and stores them in the file SECURITY.INI. The information in this file is also shared with P&E's In-Circuit Debugger and In-Circuit Simulator Software. This allows the user to reset the device and still have access to the monitor mode.

Sometimes the case comes up where the software can't pass security mode. The Target Connection and Security Dialog section has a "STATUS" section which describes the different failures and what to check in each case.

The most common reasons for not passing security are:

- You are not choosing the proper security code to pass security.
- On a power on reset, the device is not powering down to below 0.1 volts. With a Class I board (ICS with processor), you may be driving the pins on the emulation header while the device is being powered down. This back-drives current through the ports and doesn't let the device fully power down. On other classes of boards, when prompted to power down the device, the supply voltage might not be dropping lower than 0.1v which it must to have a power-on reset.
- Make sure the "Target hardware type" is set to the proper class of hardware.

There are several ways you can specify the proper security bytes:

- If you know the programmed security bytes, i.e. the bytes from $FFF6-$FFFD, you can enter them in the edit box listed "User:" and click OK(Retry).
- You can use the "Load from S19" to specify the s-record file which contains the object information currently programmed into the MCU. P&E's software will automatically extract the security information from this file and use it to pass security. Once you have specified the s-record file, click the OK(Retry) button.

- You can erase the device. Run the PROG08SZ application, and when the above box appears, select the "IGNORE security failure…" option and click OK. Use the Choose Module command to select the appropriate programming algorithm, and select Erase Module. This should erase the device. You will have to execute the Choose Module command again before you can access the blank device. Note: on some older revisions of silicon, you can't ignore the security failure, and it will bring this box back up every time you click OK(Retry). If this is the case, you should obtain the latest silicon revision from Motorola.