CMS-8GP32

A Motorola MC68HC908GP32 Microcontroller Board

xiom
anufacturing

2813 Industrial Ln. • Garland, TX 75041 • (972) 926-9303 FAX (972) 926-6063
email: sales@axman.com • web: http://www.axman.com

CONTENTS

FEATURES

The CMS-8GP32 is a Motorola MC68HC908GP32 Microcontroller development board with I/O pin headers, oscillator circuit, power supply, MON08 Debug Interface, and RS232 COM port. The MC68HC908GP32 microcontroller has a host of features which makes it ideal for low-cost embedded control applications. The CMS-08GP32 board provides a low cost and easy to use platform to develop applications on or to use in prototype generation.

Board Features

- RS232 serial port with DB9 connector from HC08 SCI port.
- Full featured MON08 port with DB9 and RS232 Levels.
- HC08 Single Chip Operation
- All HC08 I/O to Header Connector next to proto area.
- 9.8304MHz Reference Oscillator. 32.768 Crystal Oscillator.
- +6-20V Input Power Supply
- Solderless Prototype Area
- MODE Switch for easy configuration.
- 5V operation standard, 3.3V operation optional

MC68HC908GP32 Features

- CPU08 core processor
- Static to 8MHz internal Bus Operation, software selectable
- 32K Bytes Flash EEPROM on chip
- 512 Bytes SRAM on chip
- SCI serial port
- SPI serial port
- 8 channel 8 bit ADC
- Two 16 bit timers with PWM
- Keyboard Interrupt on 8 pins
- Low power modes with POR Reset and LVI functions.
- Break Point module
- MON08 development monitor
- 33 I/O pins, 26 multiplexed with internal peripherals

Development Software Features

- MCU-EZ for the HC08 provided. Contains Assembler and Linker with Windows IDE.
- Axiom MON08 Debugger and Programming Interface for Windows provided.
- DriveWay™ peripheral driver source code generator for Windows provided.
- Compatible with *HIWARE* Hiwave08 ™ software.
- Compatible with PE Micro ICS08GPZ ™ software.

GETTING STARTED

See the README.TXT file on the CD provided with this board for all the documentation and support files provided.

To get started quickly, perform the following test now to make sure the board is working correctly and you are able to communicate with your PC serial COM port:

1. Connect one end of the supplied 9-pin serial cable to a free COM port on your PC. Connect the other end of the cable to the COM port connector on the CMS-08GP32 board.
2. Run a standard ANSI terminal communications program set to 9600 baud, N,8,1. Any terminal program will work, including the simple terminal (HyperTerminal) that comes with MS Windows. See Terminal Settings in this manual for more information. You can also use the AXIOM IDE (AX-IDE) for the Terminal.
3. Verify all MODE switches are OFF, JP1 and JP2 are installed and JP3 installed on 2-3.
4. Apply power to the board by plugging in the 9V power transformer wall plug that came with the system.
5. If everything is working properly, you should see the Axiom prompt on your terminal, similar to the following:

```
AXIOM MFG. CMS-08GP32 UTILITIES 1.0
Press key 1 or 2

1. Serial Port Test    2. I/O Port Test (all switches off)
```

Your board is now ready to use. If you do not see the menu and you have not reprogrammed the HC08 flash previously, follow the steps in Troubleshooting to find why your PC COM port will not communicate with the board.

Software Development

Software development will require the use of an HC08 assembler or compiler and a host PC operating a MON08 debug interface. Supplied with this board is the Motorola MCUez HC08 assembler tools along with the Axiom MON08 IDE for Windows for Debugging and Flash programming.

In addition, the *HIWARE* Panta ™ demo CD provided contains a C compiler, assembler, linker, simulator and MON08 support tools. The *HIWARE* software will operate in “demo” mode which provides size limited assembly operation until it is registered.

The PE Micro ™ ICS08GPZ tool kit for the 68HC08 has also been tested and works with the CMS-8GP32 board. You can download it from either Motorola (http://mcu.motsps.com/) or PE Micro (http://www.pemicro.com) web sites.

A powerful source code generation tool called DriveWay™ is also provided on the support CD. This can generate C source code for the HC08 microcontroller peripherals for you, based on your specifications. See it’s readme.txt file for more information.

TUTORIAL

This section was written to help you get started with the specifics of the CMS-8GP32 software development process. Be sure to read the rest of this manual for further information. Also, you should see the 68HC08 reference manual and other documentation on the CD.

The following sections take you through the complete development cycle of a simple "hello world" program, which sends the string "Hello World" to the serial port.

Building Software

You can write source code for the CMS-8GP32 board using any language that compiles to Motorola 68HC08 instructions. Included on the CD is a free assembler provided by Motorola called MCUez. This is a Windows development environment and you should install it now if you haven't already. See the Motorola web sit to download the latest version.

You should write your source code using a standard ASCII text editor. Many powerful code editors are available or you can use the free EDIT or NOTEPAD programs that come with your computer. Once your source code is written and saved to a file, you can assemble or compile it to a Motorola S-Record (hex) format. This type of output file usually has a .MOT, .HEX, SX or .S19 file extension and is in a format that can be read by programming utilities to write the Flash Memory on the CMS-8GP32 board.

It's important to understand your development board's use of Memory and Addressing when writing source code so you can locate your code at valid addresses. You should put your program CODE somewhere in the Flash Memory area. In assembly language, you can do this with ORG statements in your source code. Any lines following an ORG statement will begin at that ORG location, which is the first number following the word ORG, for example: **`ORG $8000`**.

You must start your DATA (or variables) in a RAM location unused by your program, for example: ORG $40. You must also program the STACK register somewhere at the top of your available RAM, for example hex $23F.

A look at the example programs on the CD can make this clearer. A simple program called "`HELLO.ASM`" is provided under the \EXAMPLES directory on the CD. If you're using a compiler instead of an assembler, consult the compiler documentation for methods used to locate your code and data. To assemble the example program:

1. Launch the MCUez shell and choose the Assembler button.

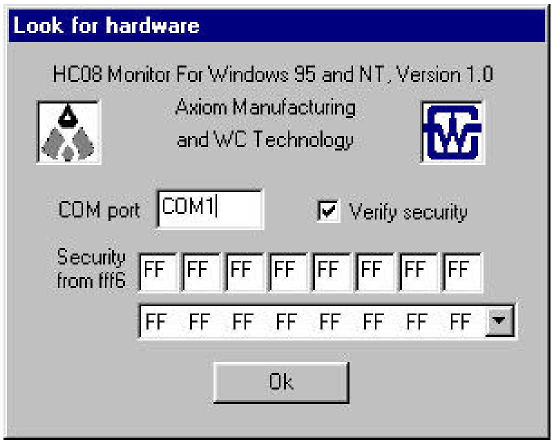
2. Chose File / Assemble from the menu and select the "HELLO.ASM" that was installed on your hard drive from the support CD. You can also just "drag" the file into the assembler window.

3. The program should assemble without any errors and produce a listing file, `HELLO.LST`, and a Motorola hex output file called `HELLO.SX`.

The listing file is especially helpful to look at when debugging your program. If your program has errors, they will be displayed and no output will be generated until you fix them.

Programming the Flash Memory

You can use the supplied MON08 utility to both program memory and debug your applications on the CMS-8GP32 board. Debugging with the MON08 is easier if you use the 2nd serial port on your PC. The following tutorial, however, will assume only 1 serial port connection is used.



COM port: COM1

☑ Verify security

Security from fff6: FF FF FF FF FF FF FF FF

FF FF FF FF FF FF FF FF

1. Disconnect power from the CMS-8GP32 board.

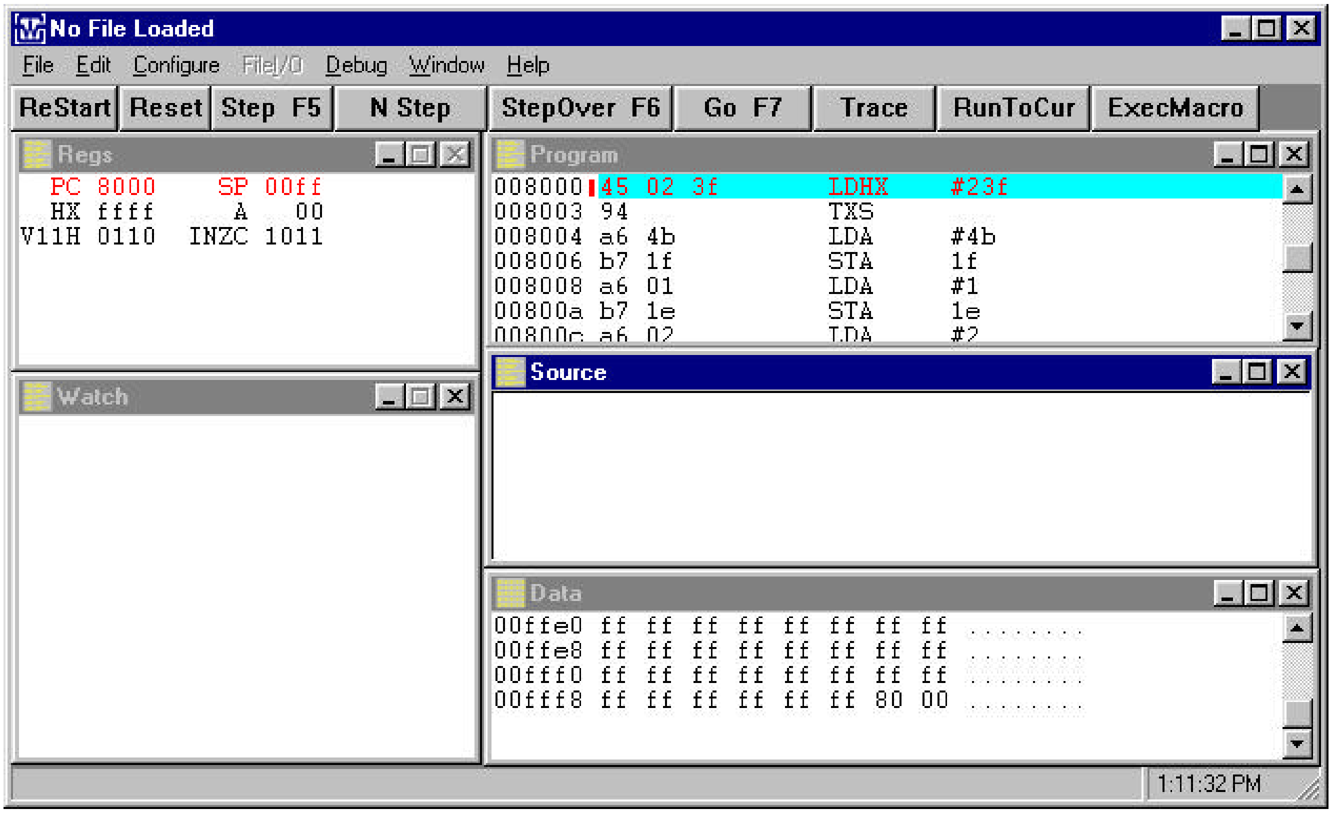
2. Move all MODE switches ON except 7 OFF. JP1 and JP2 should be installed and JP3 on 1-2.

3. Attach the serial cable to the DEBUG port connector on the CMS-08GP32 board.

4. Re-apply power to the board. NOTE: Power should remain off at least 3 seconds after the MODE switches are changed, to allow full power discharge.

5. Launch the MON08 for Windows utility installed from the support CD. In the window that comes up enter your PC COM port used to connect to the board, the "check security" box should be checked, and the security bytes can be anything (all FF for example).

6. Select the OK button. The MON08 application window (shown below) should be displayed.



```
  PC 8000     SP 00ff
  HX ffff      A   00
V11H 0110   INZC 1011
```

```
008000 45 02 3f        LDHX     #23f
008003 94              TXS
008004 a6 4b           LDA      #4b
008006 b7 1f           STA      1f
008008 a6 01           LDA      #1
00800a b7 1e           STA      1e
00800c a6 02           LDA      #2
```

```
00ffe0 ff ff ff ff ff ff ff ff  ........
00ffe8 ff ff ff ff ff ff ff ff  ........
00fff0 ff ff ff ff ff ff ff ff  ........
00fff8 ff ff ff ff ff ff 80 00  ........
```

7. Select the Debug menu item and choose “Mass erase flash EEPROM”. This will erase anything currently in Flash Memory.

8. Select the Debug menu again and choose “Program flash EEPROM”. Choose the “HELLO.SX” file you compiled in the previous section then select the Start to Program button.

When finished programming the flash EEPROM, the debugger should locate the program counter at the beginning of your program. If it’s not, right click on the Program window and go there. You can debug or execute your program now, using the single step, breakpoint, trace, watch window and other features. **NOTE**: If you want to use the Break feature, you must select “Program IRQ vector” from the Debug window. See the online Help for more information.

Executing an Application

The example “hello” program sends a string out the serial port. If you have only 1 serial cable connected to the board, you will have to test it stand-alone - outside of the MON08 debugger, since it also uses the serial port.

To make your program run on power-up, do the following:

1. Close the MON08 program.

2. Disconnect power from the board.

3. Move all MODE switches to the OFF position. (power should remain off at least 3 seconds after moving the MODE switches).

4. Move the serial cable from the DEBUG connector to the COM connector. (this is not required if you’re using 2 serial cables).

5. Start a serial terminal program on your PC (the supplied AxIDE program on the CD will work) so you can see the program run.

6. Reconnect power to the board.

You should see the message “`Hello World`” in the terminal window, every time you press a key. You can modify this simple program to display other strings or do anything you want. The procedures for assembling your code, programming memory and executing it remain the same.

MEMORY MAP

The memory map for the CMS-08GP32 is fixed by the host processor - the MC68HC908GP32. Refer to the 9GP32R3.PDF file, the MC68HC908GP32 technical manual, for details. For development use, application code under test must reside in the internal 512 byte SRAM or be programmed into the internal 32K Byte Flash Memory.

0000	**MC68HC908GP32 Registers and I/O**	64 Bytes
0040	**SRAM Memory** Stack and Program Data area	512 Bytes
0240	Unused	32K Bytes
8000	**FLASH Memory** Program Code and Constant Data area	32K Bytes
FE00	**Utility Memory** SIM and Interrupt Status, Break control, Flash control and MON08 ROM area	500 Bytes
FFF6 FFFF	**VECTORS and Security Memory** Part of Flash memory block,	10 Bytes

OPTION SWITCHES and JUMPERS

Jumpers JP1 and JP2

Option Jumpers JP1 and JP2 provide connection of the HC08 SCI TXD and RXD serial port pins to the RS232 Transceiver and COM Port. JP1 connects HC08 I/O Port E (PTE0) pin 0 (TXD) and JP2 connects HC08 Port E (PTE1) pin 1 (RXD). With the jumpers removed, the COM Port will not function as no path to the port will be provided from the HC08. The I/O pins can be used for simple I/O functions or CMOS level serial port in the users application with the jumpers removed. Connection to the HC08 pins can be made on the J2 I/O Port Header.

MODE Switch

The 8 position MODE SWITCH provides an easy method of configuring the CMS-8GP32 board operation. Mode switches 1 through 6 only effect 68HC908GP32 operation during Power On Reset condition. Mode switches should always be set prior to power being applied to the CMS-8GP32 board. For normal operation of a program installed in the 68HC908GP32 flash memory, all switches should be in the off position. MON08 operation is enabled and configured by placing one or more switches in the on position. Following is a table indicating operation of the mode switches and 68HC908GP32 I/O port pins effected. It is important that I/O ports used for MON08 configuration not be driven from user circuitry or possible misoperation or damage can occur.

MODE SWITCH	OPERATION set to ON position	HC08 I/O PORT USED
1	Enable MON08 communication path	PTA0
2	Enable MON08 for serial communication	PTA7
3	Enable Reset vector not blank operation [*1]	IRQ
4	Enable Override clock selection	PTC0
5	Enable Override clock selection	PTC1
6	Enable Override clock selection	PTC3
7	Enable MON08 BREAK detect [*1]	IRQ
8	Enable MON08 RESET	RESET

[*1] – MODE Switch position 3 and 7 should never be ON at the same time.

The CMS-8GP32 board is configured to be easy to use with 3 basic modes of operation. Modes of operation are Normal Mode, MON08 Standard Mode, and MON08 Special Mode. Following are the Mode Switch settings and capabilities for each mode of operation:

NORMAL MODE – All MODE Switches are in the off position and MON08 is disabled. Program contained in the 68HC908GP32 flash memory will be executed from Reset or Power up.

MON08 STANDARD MODE – MODE Switches 1 and 2 are ON during Power up, the 68HC908GP32 flash memory Reset Vector location (0xFFFE – FFFF) is erased (value = 0xFFFF), and the security byte values are known (values in 0xFFF6 – FFFD). MON08 will operate normally at 9600 baud with host software. Mode switches 7 and 8 can also be enabled to support host BREAK and RESET functions, however the security bytes for the IRQ vector at 0xFFFA and FFFB will not be blank and must be known (see tutorials).

MON08 SPECIAL MODE – Mode switches 1, 2, 3, 4, 5, 6, and 8 are ON during Power up. The 68HC908GP32 flash memory contents does not need to be known. This mode will allow the erasing of the flash memory so that Standard Mode can be used. After entry into special mode the Bulk Erase of the internal flash can be performed so that a subsequent power down and restart into MON08 Standard Mode can be performed.

If the security bytes are known, then Special Mode can continue to be used for the debug session. However, the host BREAK function will not operate due to Mode Switch position 7 being in the OFF position. Note that Mode Switch position 3 and 7 can not be in the ON position at the same time. In this case the RESET vector (0xFFFE – FFFF hex) can be erased while in special Mode which will allow return to MON08 Standard Mode on the next power off – on cycle.

JP3

NOTE: *This jumper is only on REV. E and later boards*. It provides clock selection between the 9.8304 MHz clock generator and the 32.768 KHz crystal. Normally user operation will apply the 32 KHz crystal with the HC08 PLL providing the bus clock under software control. The 9.8 MHz clock oscillator is used for MON08 operations by default but may also be applied by the user.

1-2	9.8 MHz clock oscillator
2-3	32 KHz crystal oscillator w/PLL

CONNECTORS

DEBUG Port

The DEBUG Port provides an RS-232 level serial interface for the MC68HC908GP32 - MON08 monitor. See the MODE Switch section for enabling and disabling the MON08 monitor. Following are the DEBUG port connections:

Pin 1, 4 & 6	Connected together for NULL of RS232 signals DSR, DTR, and CD.
Pin 2	Transmit Data to Host PC
Pin 3	Receive Data from Host PC, also provides BREAK signal, see note below.
Pin 5	Ground common
Pin 7, 8	Connected together for NULL of RS232 signals RTS and CTS, also provide RESET input from host PC RTS signal if MODE switch position 8 is ON.

BREAK Signal - The host PC can send a transmit break character for halting code execution and returning the HC08 to MON08 control from the executing application if the following conditions are met:

1. HC08 IRQ vector is enabled to perform Break service for the MON08 Monitor.
2. MODE Switch position 7 is ON (3 must be OFF).
3. The Transmitted Break character is a minimum 15ms long.

COM Port

The COM Port provides an RS-232 level interface to the MC68HC908GP32 SCI serial port for sample or user applications. See JP1 and JP2 options also. Following are the signals provided on the COM Port:

Pin 1, 4 & 6	Connected together for NULL of RS232 signals DSR, DTR, and CD.
Pin 2	SCI Transmit Data to Host terminal
Pin 3	SCI Receive Data from Host terminal
Pin 5	Ground common
Pin 7, 8	Connected together for NULL of RS232 signals RTS and CTS.

NOTE: Pins 1, 4, 6, 7, 8, and 9 have user connection pads that allow additional user application specific configuration of the COM Port such as handshake signaling. The pins can be isolated from each other by cutting the tie trace located under the pin identification print on top of the board.

- JP1 - enables connection of HC08 SCI or Port **PE0/TXD** pin to COM Port RS232 transceiver.
- JP2 - enables connection of HC08 SCI or Port **PE1/RXD** pin to COM Port RS232 transceiver.

J1 Power Connector

These 3 connection points allow for alternate power input (such as a battery or power supply) as well as output voltage supply.

+V	+7 to 16V DC unregulated Input or Output
GND	Ground
VCC	+5V Input or Output

SPECIFICATIONS

Oscillator	9.8304MHz Crystal Standard, 4.9MHz or 32KHz user option.
Maximum Clock	8MHz Internal
Operating Temperature	0 to 70 degrees C
Board Size	4.0 x 4.0 Inches
Power Requirement	6 – 20VDC @ 20ma Input Typical