DOC-0266-010-A

CMD-12A4

Development Board for the Freescale MC68HC812A4

xiom
anufacturing
M

1226 Exchange Dr. • Richardson, TX 75081 • (972) 437-3737 FAX (972) 437-3736
email: sales@axman.com • http://www.axman.com

CONTENTS

GETTING STARTED

The Axiom CMD-12A4 single board computer is a fully assembled, fully functional development system for the Freescale MC68HC812A4 Microcontroller, complete with wall plug style power supply and serial cable. To get started quickly, perform the following test now to make sure everything is working correctly:

This development board applies option selection jumpers. Terminology for application of the option jumpers is as follows:

> Jumper on, in, or installed = jumper is a plastic shunt that fits across 2 pins and the shunt is installed so that the 2 pins are connected with the shunt.
>
> Jumper off, out, or idle = jumper or shunt is installed so that only 1 pin holds the shunt, no 2 pins are connected, or jumper is removed. It is recommended that the jumpers be idled by installing on 1 pin so they will not be lost.

Development board users should also be familiar with the hardware and software operation of the target HC12 device, refer to the Freescale User Guide for the 68HC812A4 and the CPU12 Reference Manual for details. The development board purpose is to assist the user in quickly developing an application with a known working environment or to provide an evaluation platform for the target HC12. Users should be familiar with memory mapping, memory types, and embedded software design for the fastest successful application development.

Application development maybe performed by applying the preloaded Debug12 (**default**) firmware monitor or by applying a BDM cable with supporting host software. The Debug12 monitor provides an effective debug method for assembly level software, but has limitations in C code developments. For C/C++ code development it is recommended that source code or symbolic debug capability be provided in the debugging environment. The BDM interface with supporting software tools should be applied for C/C++ code development so the host PC can provide the symbolic support needed. User should verify the BDM development environment supports the C compiler to be applied, not all development environments support all compilers.

The Debug12 monitor is provided in the development board External ROM memory bank. HC12 internal ram memory and some HC12 resources are applied for monitor operation. See the Debug12 chapter for monitor details on operation and resources applied. User should note that the monitor applies operation of the HC12 external data bus in expanded wide mode on HC12 I/O ports A, B, C, D, and E for access to the external memory banks. The external ram provides development memory where code to be debugged can be loaded or modified quickly and software breakpoints applied. After the application is tested, the code can be relocated to the Program or ROM memory space of the HC12 and programmed into the external ROM memory for dedicated operation.

User applications developed by applying Debug12 monitor can be modified and relocated for operation as a stand-alone application. By applying the example DB12STRT.ASM file with user application initialization, the user application will operate from Reset or Power on conditions to provide a dedicated operation of the application. See the Tutorial Section in this manual for more information.

Follow the steps in this section to get started quickly and verify everything is working correctly.

Installing the Software

1. Insert the Axiom 68HC12 support CD in your PC. If the setup program does not start, run the file called "SETUP.EXE" on the disk.

2. Follow the instructions on screen to install the support software onto your PC. Minimum install is the AxIDE for Windows software. Not all files on the CD will install so the CD should be browsed for the documents and examples related to the CMD12A4 board or A4 type boards.

3. The utility software “AxIDE” provides a terminal program to communicate with the CMD12A4 board by serial COM port connection. The software will also provide a BUILD of source code with the AS12 assembler, PROGRAM the board ROM (EEprom) memory, and READ of linear memory contents. The Target window on the AxIDE tool bar requires the development board to be selected for correct tool button operation, CMD12A in this case.

Board Startup

Follow these steps to connect and power on the board for the default DEBUG12 Monitor operation. This assumes you're using the provided AxIDE utility (installed in the previous section) or a similar communications terminal program on your PC. Verify communication settings with the AxIDE “√” tool button. Default settings are 9600 baud, No parity, 8 data bits, and 1 stop bit. NO flow control or handshake is provided from the CMD12A4 board, these should be disabled.

1. Set the CMD12A4 board options to Default:

   MODE SWITCH all positions OFF for default monitor operation.

   CMD12A4 board option jumper **RXD0** is **ON** both pins.

   ROM_SEL option positions 1 and 8 are installed, position 10 is idle.

   RAM_SEL option positions 1, 8, and 10 are installed on both pins.

   MEM_SEL options 1, 3, 5, and 7 are installed on both pins. Position 2, 4, and 6 idle.

2. 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 COM1 port on the CMD-12A4 board.

3. Apply power to the board by plugging in the power adapter that came with the system.

4. If everything is working properly the DEBUG12 prompt message will appear in the AxIDE terminal window. AXIDE will provide a flashing cursor in the terminal window if it is the active window for keyboard entry. If the cursor is not present, click the left mouse button in the terminal window to make it active or select the window with Windows keyboard operations. Type “Help<enter” where enter is pressing the keyboard enter key and the DEBUG12 command menu will be presented.

5. Your board is now ready to use! If you do not see this message prompt, or if the text is garbage, see the **TROUBLESHOOTING** section at the end of this manual.

Support Software

There are many programs and documents on the included HC12 support CD compatible with the CMD12A4 board. Install support software from the main setup menu then browse the disk to the CMD12A4 documentation, data sheets, and examples and copy to the hard drive.

Also on the disk are free assemblers AS12 and MCU-EZ, the open source GNU C/C++ compiler tools for HC11/12, example source code, and other useful software. The introductory tutorial in this manual uses the free AS12 assembler integrated into the AxIDE program. This is a simple assembler with limited capability. For a more powerful assembly tool, install the Freescale MCUez program from the CD. This will allow you to use PAGED program memory in your application.

Software Development

Software development on the CMD-12A4 can be performed using the DEBUG12 monitor utility installed in ROM memory bank (EEPROM in sockets U10/11), a third party debugger (NoICE, CodeWarrior, etc.), or a Background Debug Module (BDM) connected to the BDM PORT connector. Any of these tools can be used to assist in creating and debugging your program.

After satisfactory operation running under a debugger, the program can be loaded into the external ROM memory bank by programming it in with the AxIDE PROGRAM operation. The user application program may then run automatically whenever the board is powered on or RESET is applied.

By default, the CMD-12A4 ships with EEPROM in sockets U10/U11 and RAM in sockets U8/U9. For debugging under DEBUG12, your program should locate itself in RAM above the internal registers and memory, for example $2000 hex.

A BDM application does not require the Debug12 monitor and has full access to all resources. In this case the application code can be applied directly in the ROM memory bank, for example at $8000 hex. The RAM in the RAM memory bank (U8/U9) can be applied in the application.

After satisfactory operation running under Debug12, the application program can be written to ROM memory (EEPROM) by relocating the start address to ROM space, $8000 for example, then selecting “Program – External EEprom” from the AxIDE utility program. When programming is complete your program will run automatically when the CMD12-A4 is powered on or RESET is applied.

To return to debug mode, you can re-program the DBUG12 monitor file: **`AX-DB12.S19`** into the board or simply re-connect the BDM.

TUTORIAL

This section should help you get started with the specifics of the CMD-12A4 development process. Be sure to read the rest of this manual as well as the documentation on the disk.

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.

Creating Source Code

You can write source code for the CMD-12A4 board using any language that compiles to Freescale 68HC12 instructions. The free Assembler AS12 which is integrated with the AxIDE program is applied here.

Write the source code using any ASCII text editor. The free EDIT, WordPad, or Notepad programs that come with a computer can be applied but the format must be basic TEXT, no rich text or document type is allowed. The source file must be simple ASCII text without any document formatting added. Once the source code is written and saved to a file, assemble or compile it to a Freescale S-Record (hex) format. This type of output file usually has a .MOT, .HEX or .S19 file extension and is in a format (S record) that can be read by the programming utilities and programmed into the CMD-12A4 board.

The BUILD button on the AxIDE tool bar will assemble programs with the AS12 program. See the AS12.html file for AS12 operation help. Limitations using AS12:

AS12 is DOS compatible. All directory / folder / file paths and names must be 8 characters maximum with no spaces or special characters. The file extension .ASM should be applied for a standard source file in assembly mnemonics.

AS12 provides a 64K byte linear memory map maximum code size. It will not produce linked or paged code. See the MCUEZ assembler for higher end features.

It is important to understand the development board's use of Memory and Addressing when writing source code so code can be located at valid addresses. For example, when in debug mode (Debug12), the application program CODE is located in External RAM memory. With assembly language the code is located in memory with ORG statements in the source code to define memory location. Any lines following an ORG statement will begin at that ORG location, which is the first number following the word ORG, for example: **`ORG $2000`**. DATA (or variables) must be defined separately and placed in a RAM memory location that is not unused by the application program section, for example: **`ORG $800`** (internal HC812A4 ram).

In “debug mode” the Debug12 monitor will handle initialization, interrupt vectors (reset, timers, etc), and the STACK. When finished debugging, initialization (SCI serial port, stack pointer) and vector tables must be added to the application to perform the monitor operations. See the DB12STRT.ASM file for a sample initialization performed by Debug12. Typically set the stack at the top of the internal RAM, in assembly this would be **`LDS #$C00`**. Also install the RESET or start up vector address in the Vector space at $FFFE/F.

If applying a software development tool that also provides a BDM cable interface to the board, the monitor installed in the ROM memory is not required. The BDM software tools may have the capability to erase and program the EEprom memory. If this is the case, develop code in the external memory without applying the monitor resources. The AX-DB12.S19 S record is provided on the support CD and in the AxIDE installation directory to program into the ROM memory bank if desired. The BDM will allow locating programs in memory and applying resources reserved for the monitor.

A look at the example programs on the disk can make all of this clearer. If you're using a compiler instead of an assembler, consult the compiler documentation for methods used to locate (MAP) the code, data and stack.

Assembling source code

An example program called “HELLO.ASM” is provided under the **`\EXAMPLES\A4`** directory of the CD and if AxIDE is installed, in the installation **`\EXAMPLE\HC12\A4`** directory. Use the correct example for the MCU type of board applied.

Assemble source code by using the AxIDE "BUILD" button or by command line tools under a DOS prompt by typing:

```
AS12 HELLO.ASM –L HELLO
```

Most compilers and assemblers allow many command line options so using a MAKE utility or batch file is recommended if command line method is applied. Run AS12 without any arguments to see all the options, or see the AS12.TXT or HTML file on the disk.

The utility software, AxIDE, provided with this board contains a simple interface to this assembler. Use it by selecting the "**Build**" button or from the Tools – Assemble menu. A prompt for the file to be assembled will be presented and the last file assembled will be shown. **NOTE**: The CMD12A board must be selected on the AxIDE tool bar pull down target menu first, or it may not build correctly.

DO NOT use long path or file names (> 8 characters). The free assembler is an older DOS based tool that does not recognize them.

If there are no fatal errors in your source code, 2 output files will be created:

`HELLO.S19`	a Freescale S-Record file that can be loaded or programmed into memory
`HELLO.LST`	a common listing file which provides physical address information with resulting opcodes and operand information. Warnings and error messages are provided with a summary at the end of the file.

AxIDE will open the listing file for viewing after the BUILD or ASSEMBLE operation. The listing file is especially helpful to look at when debugging the program. If the program has errors, they will be displayed in the listing or fatal errors will prevent output from being generated. The end of the listing file generally provides a count of errors or warnings in the file.

Running your application

After creating a Freescale S-Record file , "**Upload**" it to the development board for a test run. The provided example “HELLO.ASM” was created to run from external RAM so the Debug12 Monitor is applied to test it without programming it into ROM memory.

Verify that the CMD12A4 board is connected and operating properly by following the steps under “GETTING STARTED” until the Debug12 prompt is displayed, then follow these steps to run the Hello program:

1. Press and release the RESET button on the CMD12A4 board. The Debug12 monitor prompt will appear.

2. Type **`LOAD`** ↵
   This will prepare Debug12 to receive a S record file of a program. If you do not send a file, the CMD12A4 board will need RESET again to get the monitor prompt.

3. Select AxIDE **Upload** and when prompted for a file name select the assembled program file in s-record format that was created in the previous section called: **`HELLO.S19`**
   The S record file will be sent to the board through the serial port.

4. When finished loading the Done message will appear and the > command prompt again. Type **`G 2000`** ↵
   This tells Debug12 to execute the program at address $2000 hex, which is the start of the Hello program.

5. If everything is working properly, the message “Hello World” will be sent to the AxIDE terminal screen. Press ABORT or RESET to return to the monitor command prompt.

6. If you do not get this message, see the **Getting Started** and **TROUBLESHOOTING** section in this manual.

The hello program can be modified to display other strings or do anything wanted. The procedures for assembling code, uploading to the board, and executing remain the same. Debug12 has many features such as breakpoints, memory dump and modify, and program tracing. Type HELP at the Debug12 prompt for a listing of commands or consult the Debug12 documentation on the disk for more information.

Programming CMD12A4 ROM memory (EEprom)

After debugging, the application program can be moved into ROM Memory so it will execute automatically when power is applied to the board or a RESET is issued as follows:

1. Save a backup copy of the HELLO.ASM file then use the text editor (Notepad) to modify it as described.

2. Change the program location to ROM or Program space: Remove the comment '**;**' character before the following line to change the code location:

   ```
   ORG      $8000
   ```

   Add a “;” semicolon in front of this line to comment out the debug location:

   ```
   ;ORG           $2000
   ```

3. Remove the comment '**;**' character before the following line to initialize the stack pointer which is necessary when running outside of a debugger:

   ```
   LDS     #$C00   ; initialize stack location
   ```

4. Remove the comment '**;**' character before the following lines at the end of the file to set the Reset Vector which is necessary when running outside of a debugger:

   ```
   org     $fffe                  reset vector
   fdb     START
   ```

5. Save the new file and BUILD / Re-Assemble HELLO.ASM as described in the "Assembling Source Code" section.

6. Select the **Program** button or Tools – Program from the AxIDE menu and follow the message prompts. When prompted for a file name, select the new HELLO.S19 file. Note that the Program External (EEprom or flash) option should be checked also. Select OK when ready.

7. The AxIDE start programming window will appear now. The window prompt will indicate MODE switch positions 1, 2, and 4 are ON with 3 and 5 OFF. The ROM_SEL option jumper position 10 must be installed on both pins also. The data bus width option in the AxIDE Program window should be 16 bits for this tutorial operation.

8. Press the RESET button on the CMD12A4 board to begin the programming operation. Note that this operation will replace the Debug12 monitor and it will need to be installed again using this same procedure.

9. When finished programming: Remove or Idle ROM_SEL option position 10 and set all MODE switches OFF.

10. Press the Reset switch on the CMD12A4 board or re-apply Power to the board. The Hello program should start automatically and the “Hello World” prompt should be displayed in the terminal window. Press the RESET button again for a new Hello message.

To return to the Debug12 monitor program, start at step 6 above, select the AX-DB12.S19 file in the AxIDE examples\HC12\A4 directory, and perform steps 7, 8, and 9 to install the monitor again. the monitor prompt will appear in step 10.

Debug of 68HC12 software with the CMD12-A4 board is possible with any of the following:

- PCs serial port and the DEBUG12 monitor utility with AxIDE software.
- PCs Parallel port and a parallel type Background Debug Module (BDM) connected to the DEBUG connector on the board. BDM hosting software is required.
- PCs USB port and a USB type Background Debug Module (BDM) connected to the DEBUG connector on the board. BDM hosting software is required.
- NOICE or other monitor kernel software and hosting software.

Using the Axiom BDM

If you purchased the Axiom USB-BDM, the easiest way to get started is to become familiar with the BDM software. If you have not done so already, install the BDM software now.

After installing and running the BDM software, you must **Configure** the target it to work properly with the CMD12-A4 board. Select the menu item **Configure – Target** to see the configuration dialog box. Make sure the Device select is set to **12A4, 8000KHz clock, Expanded Wide Mode, and software breakpoint.**

Also the reset Macro file **A4RESET.MAC** should be selected in the **Configure – Macros** as the macro file after Reset. The macro file will enable the external bus controls and chip selects for loading code.

The CMD12A4 board options should be changed for default for BDM application. The MODE switch positions 1, 2, and 4 = ON for BDM or Programming. ROM_SEL position 10 should be installed also to load Program / ROM memory space.

Click the RESET button in the BDM software so the configuration changes will take effect. The memory will most likely change and the program counter will reset to whatever setting was configured or the value in external memory location $FFFE (Reset Vector).

DEBUG12 MONITOR

The Debug/Monitor software provides an embedded means for uploading your application software to RAM and executing it in a controlled environment using software breakpoints, trace and memory monitoring features. The Debug software itself resides in the on-board EEPROM devices shipped from the factory. The debug/monitor shipped with this board was originally written by Freescale and modified by Axiom and is called DBug12. The code for this program is available on the HC12 disk \Examples\A4 file called **`AX-DB12.S19`**.

Unfortunately, the D-Bug12 software resides at address $A000-$FFFF starting in the Program Memory Expansion Window Page and is too large to allow complete use of the program page window.

To execute the Debug/Monitor software the MODE switch must be in a Normal Run Mode (see Operating Modes). For most users this will mean all MODE switches are in the default off position.

Debug12 applies HC12A4 resources. It will initialize the SCI0 serial port (COM1), set the Stack pointer, block access to the interrupt vector table at $FFD6 – FFFF, and reserve internal ram from $A00 - $C00 for its own use. Installing interrupt vectors must be performed with the procedure outlined in application note AN1280 – Applying and expanding Debug12.

Debug12 commands are provided when you type HELP at the > prompt. If you are familiar with monitor software, such as the Buffalo monitor for HC11, this should be very familiar.

Command	**Parameters**	**Description**
ASM	<Address>	Assemble/Disassemble
	<CR>	Disassemble next instruction
	<.>	Exit assembly/disassembly
BAUD	<baudrate>	Set communications rate for the terminal
BF	<StartAddress> <EndAddress> [<data>]	Fill memory with data
BR	[<Address>]	Set/Display user breakpoints
BULK		Erase entire on-chip EEPROM contents
CALL	[<Address>]	Call user subroutine at <Address>
G	[<Address>]	Begin/continue execution of user code
GT	<Address>	Execute user code to address (temp breakpoimt)
HELP		Display this D-Bug12 command summary
LOAD	[<AddressOffset>]	Load S-Records into memory
MD	<StartAddress> [<EndAddress>]	Memory Display Bytes
MDW	<StartAddress> [<EndAddress>]	Memory Display Words
MM	<StartAddress>	Modify Memory Bytes
	<CR>	Examine/Modify next location
	</> or <=>	Examine/Modify same location
	<^> or <->	Examine/Modify previous location
	<.>	Exit Modify Memory command
MMW	<StartAddress>	Modify Memory Words (same subcommands as MM)
MOVE	<StartAddress> <EndAddress> <DestAddress>	Move a block of memory
NOBR	[<address>]	Remove One/All Breakpoint(s)
RD		Display all CPU registers
RM		Modify CPU Register Contents
T	[<count>]	Trace <count> instructions
UPLOAD	<StartAddress> <EndAddress>	Move a b Dump Memory in S19 record format
VERF	[<AddressOffset>]	Verify S-Records against memory contents
PC SP X Y A B D	<Register Value>	Set register contents
S XM H IM N Z V C	<CCR Status>	Set Status Bit (1 or 0)

MEMORY MAP

Following is the **DEFAULT EXPANDED WIDE** memory map for this development board:

FFFF FFFE	RESET Vector CSP0 Chip Select	2 Bytes	
FFFD C000	CMD-12A4 ROM SPACE (U10/11)	16K	
BFFF 8000	Program Memory Expansion Window Page (U10/11)	16K	Program Page Window
7FFF 7000	Data Memory Expansion Window Page (U8/9, CSD chip select)	4K	
6FFF 2000	CMD12-A4 RAM SPACE (U8/9) CSD Chip Select	20K	
1FFF 1F80	Axiom Bootload Kernel (A4 internal EEprom)		
1F7F 1000	HC12A4 Internal EEPROM Program or Data	4K	
0FFF 0C00	CMD-12A4 RAM Space (U8/9) -CSD chip select	1K	
0BFF 0800	HC12 Internal RAM	1K	
07FF 0400	CMD-12A4 RAM Space (U8/9) CSD or CS3 EPAGE if used	1K	
03FF 0200	CS0, CS1, CS2, CS3 if used CMD-12A4 utility chip selects	512 Bytes	
01FF 0000	68HC12A4 Internal Registers See HC12A4 Technical Summary (MC68HC812A4) for complete listing and usage information	512 Bytes	

HARDWARE

FEATURES and OPERATION

Y1 Crystal Oscillator

Y1 is 16.00MHz standard. This provides an instruction clock / bus speed of 8mhz to the CMD-12A4.

LV1 Reset Generator

LV1 is a voltage detector that will generate an active low RESET state if VDD supply is below +4.4 VDC.

BOOTLOADER FIRMWARE

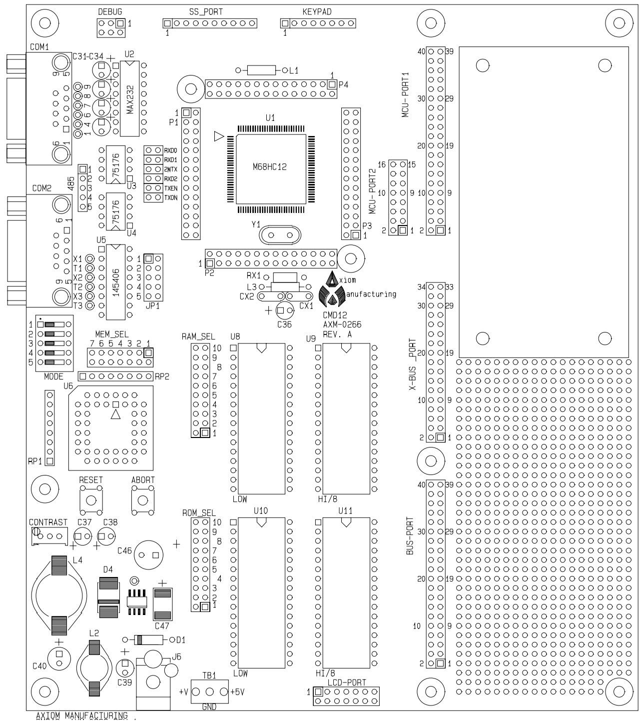
The MC68HC812A4 is pre-programmed with bootload firmware that operates in conjunction with the AxIDE Utility software to provide a low cost debugging and programming environment for the M68HC12. The 64 Byte firmware is programmed into the internal HC812A4 EEPROM and becomes operational when the 68HC812A4 is in Single Chip Mode to support utility operations. Firmware memory is mapped as follows:

Single Chip Mode (normal)	$FF80 - $FFBF and $FFFE/FFFF = Reset Vector
Expanded Modes (default)	$1F80 - $1FBF and $1FFE/1FFF = $FF80

Caution should be used when programming or erasing the internal 68HC812A4 EEPROM to assure the bootload firmware is not erased or corrupted. The EEPROT Register (Location $00F1 default) bits 0 and 1 should be set high during the users software initialization sequence to protect the firmware.

If the bootloader firmware is erased or corrupted it can be re-installed with the Monitor/Debugger operating in Normal Run Modes by loading the BOOTLOAD.S19 file and executing a G $0800 instruction. If the Monitor/Debugger has been overwritten in the on-board EEPROMs also, then a EEPROM programmer must be used to reprogram the EEPROM(s) with the program file called **`AX-DB12.S19`** to restore the Monitor operation and then reload the Bootloader as described above.

CMD-12A4 BOARD



OPERATING MODES

The CMD-12A4 allows maximum use of the features of the MC68HC12A4 by allowing the user to select the size of the data bus and memory space. This allows the user to configure the board for maximum use of the I/O lines, memory size, execution speed and other requirements. To provide these features, allow for programmability and provide a low cost development method the CMD-12A4 board uses an ON-BOARD MEMORY CONTROLLER (OBMC). The OBMC provides the necessary control signals required to program the On-Board EEPROM Memory, perform data bus size switching so both 8 and 16 bit devices can share the data bus, supply control signals to access off-board memory and determine on-board memory space size.

The operating mode of the 68HC12A4 and the CMD-12A4 OBMC are defined by the MODE SWITCH and MEM_SEL Option Jumpers. The MODE Switch defines the operating mode of the 68HC12A4 and OBMC. The MEM_SEL jumpers enable the 68HC12A4 predefined chip selects (CSP0, CSP1, CSD, CS3, and CS2) to access on-board memory via the OBMC. MEM_SEL also enables the use of address lines A20 and A21 for expanded memory space access or other 68HC12A4 features.

NORMAL RUN MODE

Normal Run Mode refers to the CMD12-A4 and the HC12 operating in a normal expanded memory access mode. MODE Switch position 4 must be OFF. In this mode the HC12/CMD12-A4 is operating via instruction and data accesses to on or off-board external memory.

PROGRAM MODE

Program Mode refers to the CMD-12A4 in programming mode with MODE Switch position 4 on and the HC12 is in Single Chip Mode with MODE Switch positions 1 and 2 on. This allows the Bootloader kernel (firmware) to operate and have access to all of memory space for reading or programming. In Programming Mode the HC12 is configured for maximum memory access (expanded memory windows enabled for CSP0 and CSD). External devices connected to the X_BUS_PORT or MCU_PORT that may affect high order address lines or data lines should be removed.

MODE SWITCH

The MODE Switch selects the HC12 operating mode and the CMD12-A4 ON-BOARD MEMORY CONTROLLER (OBMC) memory access mode.

Positions 1 & 2	HC12 MODA and MODB inputs respectively select the HC12 operating Mode. OFF places a high level and ON places a low level on the corresponding Mode pin.
Position 3	Selects CMD12-A4 On-Board memory space expansion off or on (16 or 21 bit addressing) in Normal Run Mode. Selects Programming Mode 16 or 8 bit data bus width in Programming Mode.
Position 4	Selects Normal Run Mode or Memory Programming Mode.
Position 5	Selects chip select CSP1 Swap on or off. This allows CSP1 to access On-Board RAM Space instead of On-Board Program Space for debugging purposes. Normally RAM Space is reserved for chip select CSD. Note that if Swap is on, both CSD and CSP1 access the same On-Board memory devices.

Following is a table of the **VALID** Mode Switch settings for different operations:

	1	2	3	4	5	✓ = default setting ■ = Off
✓	■	■	■	■	■	Expanded Wide (16 bit Bus), 32K Data, 32K Program Space On-Board
	■	On	■	■	■	Expanded Narrow (8 bit Bus), 32K Data, 32K Program Space On-Board
	■	■	On	■	■	Expanded Wide (16 bit Bus), 512K Data, 2M Program Space On-Board
	■	On	On	■	■	Expanded Narrow (8 bit Bus), 512K Data, 2M Program Space On-Board
	On	On	■	On	■	Single Chip, **Utility / BDM Mode** (16 bit data bus width/ U10&U11 installed)
	On	On	On	On	■	Single Chip, **Utility / BDM Mode** (8 bit data bus width/ only U11 installed)
					On	CSP1 = Swap On, Access On-Board RAM
					■	CSP1 = Swap Off, Access On-Board EEPROM

- Narrow mode uses memory devices: U9 for Data-RAM, U11 for Program-EEPROM
  U8 and U10 removed
- Wide Mode uses memory devices: U8 and U9 for Data-RAM
  U10 and U11 for Program-EEPROM

The CMD12-A4 should be RESET whenever the Mode Switch is changed.

MEM_SEL (CHIP SELECT) Option Jumpers

The Memory Select Option Jumpers enable or disable the 68HC12A4 predefined chip selects and high order address lines A20 and A21 for use by the On-Board Memory Controller.

Jumper Installed = Enabled ✓= default on

	Position	**Function**
	1	Enable address line 21
	2	Enable address line 20
✓	3	Enable CS2 – Peripheral Chip Select (128 bytes at $380 - $3FF)
	4	Enable CS3 – EPAGE Extra Page, RAM
✓	5	Enable CSD – Data
	6	Enable CSP1 – Page Chip Select
✓	7	Enable CSP0 – Must be installed for programming

- CS2 must be ON to use the LCD_PORT connector (8 bit access enabled in software).
- CSP0 must be enabled for on-board program to operate.

ROM_SEL and RAM_SEL - Memory Device Selection Jumpers

The type and size of memory devices installed on the CMD-12A4 board is configured using the RAM_SEL and ROM_SEL option jumpers. Devices installed in ROM Bank U10 and U11 sockets should be EPROM, EEPROM, or FLASH EPROM type devices. Devices installed in RAM Bank U7 and U8 should be Static RAM or EEPROM type devices. Larger memory devices in 32 pin packages may be used without memory space expansion or high order address lines enabled (Mode switch position 3). The On-Board Memory Controller will allow access to the lower addresses of the large memory device by forcing Memory Address lines MA16 thru MA21 low.

The factory setting for the jumpers should be correct for the memory devices that came with your board. If you add or modify the type or size of memory, you must change the following jumpers accordingly. All jumpers are two-pin jumpers and are installed vertically.

ROM_SEL

x **=** Jumper Installed 1 = Jumper is WRITE ENABLE. Protects EEPROM and FLASH from accidental overwrite if removed after programming.

	Device	**Size** (in bytes)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	27256	32K ROM		x						x	x	
✓	28256	32K EEPROM	x							x		1
	27512	64K ROM		x						x	x	
	29512	64K FLASH		x				1			x	
	27010	128K ROM		x						x	x	
	28010	128K EEPROM	x					1			x	
	29010	128K FLASH		x				1			x	
	27020	256K ROM		x					x		x	
	29020	256K FLASH		x				1	x		x	
	27040	512K ROM		x		x	x		x		x	
	28040	512K EEPROM	x		x			1	x		x	
	29040	512K FLASH		x	x			1	x		x	
	27080	1024K ROM		x		x	x		x		x	

RAM_SEL

x **=** Jumper Installed 1 = Jumper is WRITE ENABLE

	Device	**Size** (in bytes)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
✓	62256	32K RAM	x							x		1
	28256	32K EEPROM	x							x		1
	62010	128K RAM	x		x		x			x		1
	28010	128K EEPROM	x					1			x	
	62040	512K RAM	x		x		x		x			1
	28040	512K EEPROM	x		x			1	x		x	

You can determine the size of a memory device by reading the label on top of the chip. Memory devices that contain 256 in the part number are usually 8K byte. Those with 512 are usually 64K. If you don’t recognize the memory type you can look up the part number in a catalog or device manual. If the chip is by Atmel™ or XICOR™ it’s probably an EEPROM. If it has HY or SEC it’s probably RAM.

Address Decoding and Expanding Memory

The CMD12-A4 ships standard with 128K bytes of on board memory. The 68HC12A4 Microcontroller can access up to 4 Megabytes of program space (CSP/CSP1) memory or 1 Megabyte of data space memory (CSD or CS3 EPAGE) using special memory expansion address windows and memory page operations. The CMD12-A4 On-Board Memory Controller can also provide additional address decoding to allow on and off board memory operations. On-Board RAM (U8 and U9) space is accessed with HC12 chip select CSD normally. Access to RAM is also provided via the CS3 EPAGE or CSP1 chip select if enabled. On-board ROM (U10 and U11) space is accessed with HC12 CSP0 normally and CSP1 optionally.

The CMD12-A4 board is configured to allow memory expansion in 2 basic modes determined by MODE Switch position 3 in normal run modes. Switch 3 OFF configures the CMD12-A4 to provide 64K bytes of on-board memory and all other memory off-board. Switch position 3 ON enables memory expansion on-board up to half the available HC12 addressable memory space with 2M Byte Program space and 512K Byte of Data space.

When memory expansion is enabled on the CMD12-A4, the memory map is split in the center so that Data space (CSD or CS3 EPAGE) expands from address 0 upward to 512K (A19 low) and resides in RAM (U8 & 9). Program space (CSP0 or CSP1) expands downward to 2M byte (A21 high) from address $3F,FFFF hex and resides in ROM (U10 &11). The rest of the Data space (A19 high) and Program space (A21 low) may be accessed off board. X_BUS_PORT Control signals EXDAT and EXPRG (both active low) provide off-board address access decoding.

For example, to expand memory space the appropriate bits in the HC12 WINDEF register will enable memory paging with the HC12 chip selects. If CSD expansion is enabled with DPAGE, memory between $7000 - $7FFF is available as 256 x 4K data pages of which the lower 128 pages may be accessed on-board. The DPAGE register contains the data page number currently being accessed. Also available are 256 x 16K program pages from $8000 - $BFFF if the PPAGE is enabled. The PPAGE register contains the program page number.

Off-board memory expansion is possible via the BUS_PORT and X_BUS_PORT Connectors. Section 10 of the Freescale HC12 Reference Manual explains this technique in detail. Also see the MC68HC812A4 Technical Data book for complete configuration register and chip select information.

PORTS and CONNECTORS

Header Ring P1 - P4 Connectors

The Freescale 68HC812A4 Microcontroller is attached to four dual row 14 pin connectors (28 pins each) for direct access to I/O pins.

P1

Vss	1	**1**	**2**	2	Vdd
PJ0	3	**3**	**4**	4	PJ1
PJ2	5	**5**	**6**	6	PJ3
PJ4	7	**7**	**8**	8	PJ5
PJ6	9	**9**	**10**	10	PJ7
A16/PG0	11	**11**	**12**	12	PG1/A17
A18/PG2	13	**13**	**14**	14	Vdd
Vss	15	**15**	**16**	16	PG3/A19
A20/PG4	17	**17**	**18**	18	PG5/A21
BKGD	19	**19**	**20**	20	PD0/D0
D1/PD1	21	**21**	**22**	22	PD2/D2
D3/PD3	23	**23**	**24**	24	PD4/D4
D5/PD5	25	**25**	**26**	26	PD6/D6
D7/PD7	27	**27**	**28**	28	PC0/D8

P2

D9/PC1	29	**1**	**2**	30	PC2/D10
D11/PC3	31	**3**	**4**	32	PC4/D12
D13/PC5	33	**5**	**6**	34	PC6/D14
D15/PC7	35	**7**	**8**	36	PE0/XIRQ
IRQ/PE1	37	**9**	**10**	38	PE2/R/W
LSTR/PE3	39	**11**	**12**	40	/RESET
Vss	41	**13**	**14**	42	Vdd
Vddpll	43	**15**	**16**	44	XFC
Vsspll	45	**17**	**18**	46	EXTAL
XTAL	47	**19**	**20**	48	PE4/ECLK
MODA/PE5	49	**21**	**22**	50	PE6/MODB
ARST/PE7	51	**23**	**24**	52	PB0/A0
A1/PB1	53	**25**	**26**	54	PB2/A2
A3/PB3	55	**27**	**28**	56	PB4/A4

P3

A5/PB5	57	**1**	**2**	58	PB6/A6
A7/PB7	59	**3**	**4**	60	PA0/A8
A9/PA1	61	**5**	**6**	62	PA2/A10
A11/PA3	63	**7**	**8**	64	PA4/A12
A13/PA5	65	**9**	**10**	66	PA6/A14
A15/PA7	67	**11**	**12**	68	PF0/CS0
CS1/PF1	69	**13**	**14**	70	PF2/CS2
CS3/PF3	71	**15**	**16**	72	PF4/CSD
CSP0/PF5	73	**17**	**18**	74	PF6/CSP1
PH0	75	**19**	**20**	76	PH1
PH2	77	**21**	**22**	78	PH3
Vdd	79	**23**	**24**	80	Vss
PH4	81	**25**	**26**	82	PH5
PH6	83	**27**	**28**	84	PH7

P4

V_{RH}	85	**1**	**2**	86	V_{RL}
PAD0	87	**3**	**4**	88	PAD1
PAD2	89	**5**	**6**	90	PAD3
PAD4	91	**7**	**8**	92	PAD5
PAD6	93	**9**	**10**	94	PAD7
V_{DDA}	95	**11**	**12**	96	V_{SSA}
RxD0/PS0	97	**13**	**14**	98	PS1/TxD0
RxD1/PS2	99	**15**	**16**	100	PS3/TxD1
SDI/PS4	101	**17**	**18**	102	PS5/SDO
MOSI/PS6	103	**19**	**20**	104	PS7/SCK
PT0	105	**21**	**22**	106	PT1
PT2	107	**23**	**24**	108	PT3
PT4	109	**25**	**26**	110	PT5
PT6	111	**27**	**28**	112	PT7/PAI

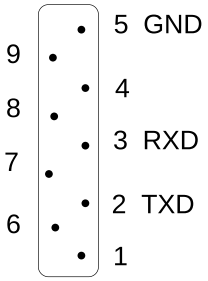
- The small numbers next to the connector pin numbers are MC68HC812A4 package pin numbers for reference.
- See the M68HC812A4 Technical Manual for more information on I/O ports.

SERIAL COM PORTS

COM1 interfaces to the HC12 internal SCI0 serial port (I/O PS0 and PS1) and is a simple three wire asynchronous serial interface with hard wired Clear to Send (CTS) and Data Terminal Ready (DTR). These two logic level signals are coupled thru an RS232 level shifter to the COM1 connector.

COM1 is the default serial interface for the Monitor/Debugger and AX12 utility software.

COM1 DB9S Style Connector

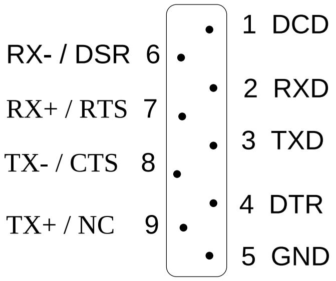


- Permanent jumpers between following pins:
  4 → 1 and 6 (DTR/DSR/DCD)
  7 → 8 (RTS/CTS)
- COM1 is set to connect directly to a PC serial port with a straight thru type of cable (supplied).

Note RXD0 enable option on the RS232 Option jumper.

COM2 interfaces to the HC12 SCI1 serial port (I/O port PS2 andPS3) and can be implemented either as an RS232 communications port or as an RS422/RS485 communications port. Additional HC12 I/O Port J lines are available to provide control and handshake signals for the serial port. Normal 422/485 interface access is provided at the 485 Header, however, these signals can be coupled to the COM2 DB9P connector with a simple modification.

COM2 DB9P Style Connector



- To connect COM2 to a PC serial port you must use a NULL modem cable or NULL modem adapter.
- Option jumpers RXD1 and RXD2 select the serial receive data source as RS232 or RS422/485 respectively. Only one jumper should be installed.
- Flow and Control signals may be provided by HC12 I/O Port J lines PJ0 to PJ3. Port J provides Interrupt capability to enhance the operation of COM2 in software. See JP1 options.
- RX-/RX+/TX-/TX+ are 422/485 signals if the option is used.

COM Port Option Jumpers

RXD0	ON selects COM1 RS232 Receive (**Default = ON for DBUG12)**
RXD1	ON selects COM2 RS232 mode Receive (RXD2 open)
2WTX	ON selects 485 mode 2 wire Transmit and Receive with PJ0 as the TX enable control. 2W Interface is 485 header pins 1 and 2.
RXD2	ON selects COM2 485 mode Receive (RXD1 Open)
TXEN	ON selects 4W Mode with PJ0 as the TX output enable control. 485 Header pins 1 and 2 are Receive and pins 3 and 4 are Transmit (Jumper TXON = open).
TXON	ON selects 4W mode with Transmit Enabled always ON, no on / off output control. (Jumper TXEN = open)

NOTE: I/O Port J0-3 must be configured and applied in software to perform the control function. Verify only one function is selected for each Port J line.

JP1 - Option Jumpers, COM2 RS232 flow and control signals

HC12 Ports J0 – J3 provide optional flow and control signals:

DTR	ON selects HC12 PJ0 as DTR control output
CTS	ON selects HC12 PJ1 as CTS control input
DSR	ON selects HC12 PJ2 as DSR control input
DCD	ON selects HC12 PJ3 as DCD control
RTS	ON selects HC12 PJ0 as RTS control output

485 HEADER

1	2W TX/RX or 4W RX B wire (-)
2	2W TX/RX or 4W RX A wire (+)
3	4W TX B wire (-)
4	4W TX A wire (+)
5	Ground

COM2 DB9P connector to RS422/485 Connection Option

To connect the COM2 DB9P connector pins 6 to 9 to the 422/485 interface the following modification can be performed:

1. Remove the U5 (MC145406) IC from its socket and store for possible future use.
2. Install shorting wires or SMT 1206 size 0 ohm resistors on the underside of the CMD12-A4 across empty component pads labeled R11, R12, R15 and R16.
3. To restore RS232 operation, remove shorting wires or components at R11, R12, R15 and R16 and re-install U5.

LCD_PORT Connector

The LCD Display interface is connected to the data bus and memory mapped to locations $3F0 thru $3F3 and controlled by CS2 which must be jumper enabled (see MEM_SEL). Due to bus speed requirements of the LCD Modules the read and write addresses of the LCD are separated. Addresses $3F0 and $3F1 are Command Write and Data Write respectfully. Addresses $3F2 and $3F3 are Command Read and Data Read respectfully. Caution must be used not to Write to the Read addresses as a bus conflict will occur.

The interface supports all OPTREX™ DMC series displays up to 80 characters and provides the most common pinout. Power, ground, and Vee are also available at the LCD_PORT connector. The potentiometer, VR1 Vee ADJUST located near the RESET Switch, is used to adjust the contrast of the LCD display by varying Vee from +5 to -5 Volts.

See the file `KEYLCD12.ASM` on the software disk for an example program using this LCD connector.

+5V	**2**	**1**	GND
MA0	**4**	**3**	VEE / Contrast
LCDCS	**6**	**5**	MA1
D9	**8**	**7**	D8
D11	**10**	**9**	D10
D13	**12**	**11**	D12
D15	**14**	**13**	D14

Address $3F0 = LCD Control Register
Address $3F1 = LCD Data Register

KEYPAD Connector

The KEYPAD connector is an eight position connector that implements HC12 Port H as a passive keypad interface. This interface is implemented as a software key scan. Pins PH0 - PH3 are columns which are reset low to read the row condition on PH4 - PH7 which are active low. See the file **`KEYLCD12.ASM`** on the software disk for an example program using this keypad connector.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
H0	H1	H2	H3	H4	H5	H6	H7

BUS and XBUS PORTS

The **BUS_PORT** and **X_BUS_PORT** support off-board memory or peripheral devices.

BUS_PORT				**X_BUS_PORT**			
/ RESET	**40**	**39**	GND				
Y7	**38**	**37**	ECLK				
Y6	**36**	**35**	R/W				
CS3	**34**	**33**	+5V	GND	**34**	**33**	XIRQ/PE0
/IRQ	**32**	**31**	Y5	GND	**32**	**31**	ARS/PE7
Y4	**30**	**29**	Y3	PE6/MB	**30**	**29**	LST/PE3
Y2	**28**	**27**	Y1	PE5/MA	**28**	**27**	+5V
Y0	**26**	**25**	/WR	PF6/CP1	**26**	**25**	CSD/PF4
MA6	**20**	**19**	MA9	/EXPRG	**20**	**19**	/EXDAT
MA7	**22**	**21**	MA8	PF1/CS1	**22**	**21**	CS0/PF0
MA13	**24**	**23**	MA12	PF5/CP	**24**	**23**	CS2/PF2
MA5	**18**	**17**	MA11	A21/PG5	**18**	**17**	SIZE
MA4	**16**	**15**	/OE	MA19	**16**	**15**	PG4/A20
MA3	**14**	**13**	MA10	MA17	**14**	**13**	MA18
MA2	**12**	**11**	MA1	MA15	**12**	**11**	MA16
D15	**10**	**9**	MA0	D7/PD7	**10**	**9**	MA14
D14	**8**	**7**	D8	D6/PD6	**8**	**7**	PD0/D0
D13	**6**	**5**	D9	D5/PD5	**6**	**5**	PD1/D1
D12	**4**	**3**	D10	D4/PD4	**4**	**3**	PD2/D2
D11	**2**	**1**	GND	D3/PD3	**2**	**1**	GND

BUS_PORT Signals

D8 - D15	High Byte Data Bus in Wide Expanded Mode, Data Bus in Narrow Expanded mode or for 8 bit accesses in Wide Mode.
MA0 – MA13	Memory Address 0 to 13, Driven by Bus Size Switch for bus addressing.
/OE	Memory Output Enable signal, Active Low. Valid with ECLK and R/W high.
Y0 - Y7	Auxiliary Chip Selects, 16 Bytes each located at $380 hex to $3FF hex driven by HC12 CS2 chip select. MEM_SEL option jumper 3 to enable.
/WR	Memory Write Enable signal, Active Low. Valid with ECLK high and R/W low.
/IRQ	HC12 IRQ (PE1) Interrupt Input.
CS3	HC12 chip select CS3 (PF3). BUS_PORT can expand to memory or peripheral devices using the CS3 chip select.
R/W	HC12 Read/Write (PE2) control signal.
ECLK	HC12 ECLK (PE4) bus clock signal. Stretch should be enabled in software.
/RESET	**CMD12-A4 active low RESET signal.**

X_BUS_PORT Signals and I/O

PD0/D0 - PD7/D7	HC12 Port D lines, Low Byte Data Bus in Wide Expanded Mode.
MA14 – MA19	Memory Address 14 to 19, Driven by Bus Size Switch for proper bus size addressing. MA16 to MA19 are forced low if not using Expanded Memory Access (see Mode Switch). PG4/A20 - PG5/A21 HC12 Port G4/5 or A20/A21 address lines respectfully.
SIZE	Bus width signal generated by OBMC (see Operating Modes). High level signal is 16 bit (Wide) data bus and low level signal is 8 bit (narrow) bus. Signal operates the on-board bus size switch.
/EXPRG	External (off-board) Program Memory Access signal. Low active signal when HC12 CSP0 or CSP1 chip select access is outside on-board ROM Space memory range.
/EXDAT	External (off-board) Data memory Access signal. Low active signal when HC12 CSD chip select is outside on-board RAM Space memory range.
CS0/PF0 - CSP1/PF6	HC12 Programmable Chip Selects or Port F I/O port.
PE3/LSTRB	HC12 Port E3 is used as the LSTRB signal to indicate bus size changes to the OBMC. User initialization software should support this operation.
PE5/MODA - PE6/MODB	HC12 Mode Select A and B or Port E5 and 6 I/O lines. These lines must be available as inputs during RESET operation to select the HC12 operating mode.
PE7/ARST	HC12 Port E7 is available as a general purpose I/O.
PE0/XIRQ	HC12 Port E0 or XIRQ interrupt input is used as the ABORT Switch while the Debug/Monitor is operating.

MCU_PORTS 1 and 2

The **MCU_PORT1 and 2** provide access to the peripheral features and I/O lines of the HC12. Note that address and data ports A,B,C, D and part of E may not be applied as I/O when the external memory bus is operating.

MCU PORT 1

A19/PG3	**40**	**39**	PG2/A18
A17/PG1	**38**	**37**	PG0/A16
PJ7	**36**	**35**	PJ6
PJ5	**34**	**33**	PJ4
PJ3	**32**	**31**	PJ2
PJ1	**30**	**29**	PJ0
PT7	**28**	**27**	PT6
PT5	**26**	**25**	PT4
PT3	**20**	**19**	PT2
PT1	**22**	**21**	PT0
PAD7	**20**	**19**	PAD6
PAD5	**18**	**17**	PAD4
PAD3	**16**	**15**	PAD2
PAD1	**14**	**13**	PAD0
V_{RL}	**12**	**11**	V_{RH}
GND	**10**	**9**	+5V
PH7	**8**	**7**	PH6
PH5	**6**	**5**	PH4
PH3	**4**	**3**	PH2
PH1	**2**	**1**	PH0

MCU PORT 2

PA7/A15	**16**	**15**	PA6/A14
PA5/A13	**14**	**13**	PA4/A12
PA3/A11	**12**	**11**	PA2/A10
PA1/A9	**10**	**9**	PA0/A8
PB7/A7	**8**	**7**	PB6/A6
PB5/A5	**6**	**5**	PB4/A4
PB3/A3	**4**	**3**	PB2/A2
PB1/A1	**2**	**1**	PB0/A0

MCU_PORT1 Signals

PH0 – 7	Port H I/O lines. Also used as the CMD12-A4 Keypad Port.
Vrh/Vrl	HC12 A/D Converter Reference Pins. See A/D Reference Section.
PAD0 – PAD7	HC12 Port AD is an input port or the A/D Converter inputs.
PT0 - PT7	HC12 Port T is a general purpose port or Timer I/O port.
PJ0 - PJ7	HC12 PortJ lines. Also used as CMD12-A4 COM2 and SS_Port control.
PG0/A16 - PG3/A19	HC12 Port G I/O lines, available if expanded memory space not used. (A16-A19)

MCU_PORT2 Signals

PA0 – 7	HC12 Port A I/O lines. Applied as Address bus signals A8 - A15 by default.
Vrh/Vrl	HC12 Port B I/O lines. Applied as Address bus signals A0 – A7 by default.

A/D Reference Lines

The VRH and VRL lines from the HC12 are connected to +5v through R2 and to ground through R1 respectively. These two surface mount resistors are on the bottom (solder) side of the CMD12-A4 board. The resistors are identified on the silk screen by their reference designators. The appropriate resistor(s) need to be removed in order to apply an external reference to the VRH and/or VRL inputs.

SS_PORT Connector

The Simple Serial connector can be used to communicate with external devices through the SPI Port features of the 68HC12. Up to five separate SPI serial devices can be supported by the SEL lines defining PJ4 - PJ7.

See the files **`SS-AD12.ASM`** and **`SS-DA12.ASM`** for example programs using this connector. Prebuilt Simple Serial devices with software drivers are available from the manufacturer.

		|-----	SPI	----|		|----	Select Lines		----|
		MISO	MOSI	SCK	SS	SEL1	SEL2	SEL3	SEL4
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
+5	GND	PS4	PS5	PS6	PS7	PJ4	PJ5	PJ6	PJ7

CMD12-A4 OTHER FEATURES

Trace Jumper

The Trace Jumper is not used at this time and should not be installed.

RESET Generation

The CMD12-A4 has a voltage level detector (U7) that generates an active low Reset when the supply voltage is below ~4.5VDC. The RESET Switch will apply a ground to the Reset line as long as the switch is depressed.

Power Supply

The CMD12-A4 has a switching regulator power supply that accepts +6 to +30VDC and outputs +5VDC at up to 500ma. Normally input voltages are supplied by the provided wall-plug installed into J6. TB1 provides access to output or optional input to the +V input voltage, Ground and +5VDC system supply. Caution should be used if supplying an external +5VDC system voltage so that a voltage level of +6VDC is not exceeded or permanent damage to components on the CMD12-A4 board may occur.

ABORT Switch

The Abort Switch applies a Ground to the HC12 XIRQ interrupt input. It is used by the Debug/Monitor to abort user programs executing under the Monitor.

PHASE-LOCKED LOOP

CMD12-A4 HC12 PLL loop filter components RX1, CX1 and CX2 are not installed. If PLL operation is desired, the user should review the Freescale Application Note and determine the component values to be used for the frequency of desired operation.

DEBUG Port

The CMD12-A4 Debug Port is a 6 pin header compatible in pinout with the Freescale BDM/Debug port specification. Axiom manufacturers a Background Debug Module for this board, the part number is **USB BDM12**.

TROUBLESHOOTING

The CMD12-A4 board is fully tested and operational before shipping. If it fails to function properly, inspect the board for obvious physical damage first. Ensure that all IC devices in sockets are properly seated.

The most common problems are improperly configured communications parameters and attempting to use the wrong COM port (on the PC AND on the development board). Verify that your communications port is working by substituting a known good serial device, or by doing a loop back diagnostic. Verify that no other devices are conflicting with the port (such as a mouse, modem, etc.).

Check your hardware configuration jumpers and switches. Verify the power source. You should measure 9 volts between GND and +9V test point pads on the board near J1. If no voltage is found, verify wallplug connections to 115VAC outlet and power connector. Disconnect all external connections to the board except for COM1 to the PC and the wall plug. Follow these steps in the order given:

Troubleshooting Steps

1. Visual Inspection
2. Verify that all MODE switches are switched OFF.
3. Verify power by checking for +9 volts between GND and +9V test point pads.
4. Verify the Monitor EEPROM for proper installation (no bent pins) and proper jumper settings for the device used.
5. Re-Check the communications parameters.
6. Disconnect any peripheral devices including display and keyboard.
7. Make sure that the RESET line is not being held low.
8. Verify presence of 8MHz sine wave on the crystal if possible.
9. Please check off these steps and any others you may have performed before calling so we can better help you.

Tips and Suggestions

Following are a number of tips, suggestions and answers to common questions that will solve most problems users have with the AXIDE support software. There also may be a newer version of the AXIDE utility software available. You can download the latest version for FREE on our web page at: WWW.AXMAN.COM

AXIDE Program

- Select the CMD12A board for the Target on the AXIDE tool bar. Follow prompts to apply utilities such as Program. DBUG12 will prompt in the terminal window by default.
- If you're trying to program memory or start the HC12 Utilities, make sure MODE switches 1, 2 and 4 are all ON and 5 is off. Switch 3 should be ON for 8-bit mode, OFF for 16-bit mode.
- If ROM_SEL jumper 10 has been removed for write protection, it must be re-installed to program external EEPROM. This is often the cause of programming checksum errors.
- If you're trying to execute a program, for example running the Debug Monitor from the Terminal window, the MODE Switches must be set to Expanded mode. Usually this means all switches OFF, however check the MODE Switch section if you're not sure.
- If every other byte you read is wrong ($FB for example) then the board is probably set to 16-bit mode and you are accessing it as 8-bit mode.
- Be certain that the data cable you're using is bi-directional and is connected securely to both the PC and the board. Also, make sure you are using the correct serial port.
- Make sure the correct power is supplied to the board. You should only use a 9 volt, 300 mA adapter or power supply. If you're using a power strip, make sure it is turned on.
- Make sure you load your code to an address space that actually exists. See the Memory Map if you're not sure.
- If the BUILD option doesn't work properly on your system verfiy the Path to the source is DOS compliant. 8 character maximum directory / folder names and file name with a 3 character extension such as "source.ASM".

Code Execution

- Make sure ALL jumpers are set correctly according to your board's configuration. Read the hardware manual section on jumpers carefully if you're not sure.
- Always remember to move the MODE switches 1,2,3 and 4 back to their correct positions after programming memory.
- If you're using D-Bug12 breakpoints are not acknowledged if you use the monitor CALL. You must use one of the GO commands instead. This will be fixed in a later version.
- If you programmed your code into EEPROM memory over the Debug Monitor and it doesn't run check the HC12 reset vector located at FFFEh - FFFFh. These 2 bytes contain the address in the HC12 where execution will begin when the unit is powered on.