

Follow these steps to connect and power on the EVB board for default dBUG monitor operation.

- 1) Connect EVB board UART0 / Terminal port with a serial cable to host PC COM port.
- 2) Load the ColdFire support CD into the PC and install the AxIDE terminal software from the utilities directory, OR configure HyperTerminal for a direct connection to the PC COM port to be applied for serial communication with the EVB board. Set the baud rate to 19.2K baud, 8 data bits, 1 stop bit, and no parity. Software XON / XOFF flow control should be enabled for flash memory support operations. Use the AxIDE '√' tool bar button to configure the COM port on the PC.
- 3) Apply power to the EVB board by installing the wall plug power supply between a wall outlet and the PWR Jack on the board. Turn ON the ON / OFF switch. The board voltage indicators should turn on at this time.
- 4) Observe the AxIDE or HyperTerminal window display for the dBUG monitor prompt. Prompt should be similar to the following:

Reset

```
ColdFire MCF5223 on M5223EVB
Firmware v4a.1a.1a (Built on March 17, 2006 15:46:51)
Copyright 2006 Freescale Semiconductor, Inc.
```

Enter 'help' for help.

dBUG>

- 5) dBUG and the EVB board is ready to use now. See the dBUG monitor manual for additional monitor information and further details on application.

dBUG Commands

dBUG provides a simple interface to apply the MCF5223x. User may view and modify MCF5223x memory and registers, load small executables, control application execution, and enter or display assembly level instructions via the monitor. Following is a list of available commands for this EVB application. Additional user input may be required depending on the command function. For commands that accept an optional <width> to modify the memory access size, the valid values are:

- B = 8-bit (byte) access
- W = 16-bit (word) access
- L = 32-bit (long) access

When no <width> option is provided, the default width is "W", 16-bit.

The core ColdFire register set is maintained by dBUG. These are listed below:

- A0 - A7
- D0 - D7
- PC
- SR

| MNEMONIC | SYNTAX | DESCRIPTION |
|-----------------|---|------------------------------------|
| ASM | asm <<addr> stmt | Assemble |
| BC | bc addr1 addr2 length | Block Compare |
| BF | bf <width> begin end data <inc> | Block Fill |
| BM | bm begin end dest | Block Move |
| BR | br addr <-r> <-c count> <-t trigger> | Breakpoint |
| BS | bs <width> begin end data | Block Search |
| DC | dc value | Data Convert |
| DI | di <addr> | Disassemble |
| DL | dl <offset> | Download Serial |
| DLDEBUG | dldbug | Download dBUG Update |
| DN | dn <-c> <-e> <-i> <-s> <-o offset> <filename> | Download Network |
| FL | fl <command> dest <src> size | Flash write or erase |
| GO | go <addr> | Execute |
| GT | gt addr | Execute To |
| HELP | help <command> | Help |
| IRMD | irmd <module.register> Internal | Internal Register Display / modify |
| LR | lr <width> addr | Loop Read |
| LW | lw <width> addr data | Loop Write |
| MD | md <width> <begin> <end> | Memory Display |
| MM | mm <width> addr <data> | Memory Modify |
| MMAP | mmap | Memory Map Display |
| RD | rd <reg> | Register Display (core) |
| RM | rm reg data | Register Modify (core) |
| RESET | reset | Reset |
| SD | sd | Stack Display (contents) |
| SET | set <option value> | Set Configurations |
| SHOW | show <option> | Show Configurations |
| STEP | step | Step (Over) |
| SYM | symbol <symp> <-a symp value> <-r symp> <-C I s> | Symbol Management |
| TRACE | trace <num> | Trace (Into) |
| UP | up begin end filename | Upload binary data |
| VER | version | Show dBUG Version |
| LEDS | leds <NR number> | EVB Led indicator demo |

dBUG Memory Map

| | |
|-------------------------|---|
| 0x00000000 > 0x0001FFFF | dBUG Flash memory space, 128K bytes |
| 0x00020000 > 0x0003FFFF | User Flash memory space, 128K bytes |
| 0x20000000 > 0x200003FF | User Interrupt vectors, 1K bytes |
| 0x20000400 > 0x20003FFF | dBUG data memory / Ram space, 15K bytes |
| 0x20004000 > 0x20007FFF | User Ram memory space, 16K bytes |
| 0x40000000 | ISP Base |