HI-WAVE

Serial Debug Interface SDI target

Product Manual	Manual Date
HI-WAVE - SDI	10/97 v2.2

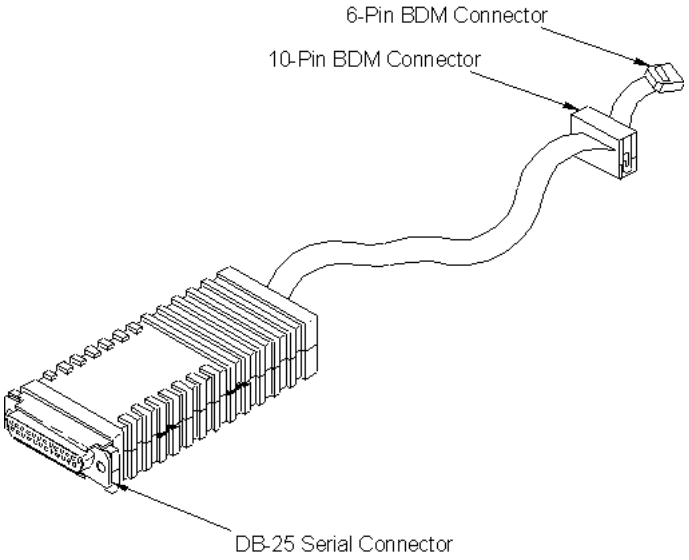
Contents

SDI Target Component

Introduction

Another advanced feature of HI-WAVE for the embedded system development world is the ability to load different Framework targets. The SDI Serial Debug Interface is introduced in this document.

The SDI is an interface developed by Motorola and used by HI-WAVE to communicate with an external system also called *target system*.



With this interface, you can download an executable program from the HI-WAVE environment to an external target system based on a Motorola MCU which will execute it. You will also have the feedback of the real target system behavior to HI-WAVE.

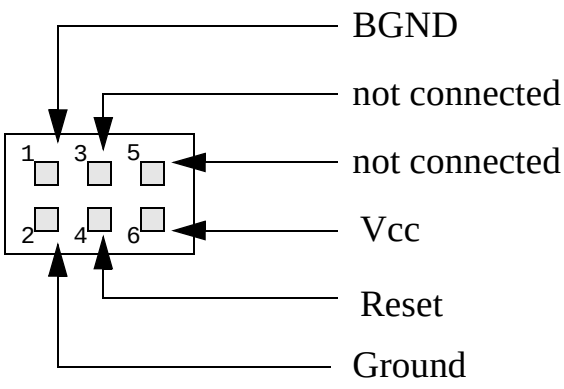
HI-WAVE will fully supervise and monitor the MCU of the target system i.e. control the CPU execution. You can read and write in internal/external memory (even when the CPU is running), single-step/run/stop the CPU, set breakpoints in the code.

> *Note:* ***Unconcerned Components*** *As the code is executed by an external MCU, memory statistics are not available with the SDI target component. Therefore, Profiling, Coverage analyzing, watchpoints and I/O simulation will not work with the SDI.*

Interfacing Your System and a Target

The SDI interface is designed around a serial communication link. It is supported by any available communication device of your system (PC or SUN). The communication protocol between the SDI and your system is fully handled by the SDI Target driver which is automatically loaded with the SDI Target Component.

The target hardware has to be equiped with a BDM connector which connects the SDI. You can set the baudrate (Please see section *SDI Default Environment*). The BDM port 6-pin connector is shown below.



The SDI - target system communication is serial. The communication protocol is defined by Motorola in the *CPU 12 Reference Manual, section 8 (Development and Debug Support).* However the user does not need to know this protocol to run an SDI debugging session.

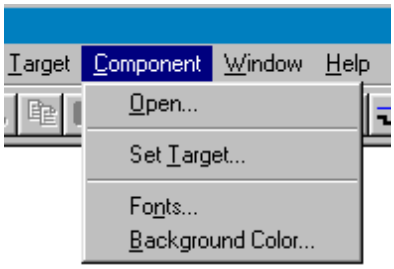
The SDI is powersupplied by the target system. This power supply must be conformed to the TTL norm. If not, the SDI should have its own supply. Please also refer to the SDI hardware manual, *SDI INTERFACE USER'S MANUAL* from Motorola.

Loading the SDI Target

Usually the target is set in the `PROJECT.INI` file, where `Target=Motosil`. The MotoSIL driver detects automatically that the SDI target is connected to your system. However, if nothing is detected, an error message pops up and inform you that the target is not connected or is connected to a different port. A *Communication Configuration* dialog pops up and you can set yourself the correct parameters (bau-

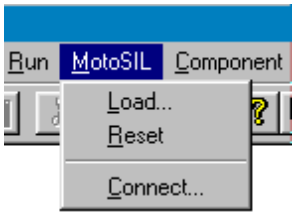
rate and communication port). Please see the following section *Communication Configuration*.

If no target is set in the `PROJECT.INI` file or if a different target is set, you can load the SDI driver selecting in the main menu *Component* | *Set Target...* as shown below and choose *MotoSIL* in the list of proposed targets.

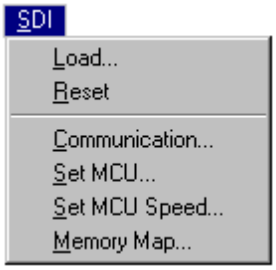


The MotoSIL driver tries automatically to find the SDI target and behaves as described above for automatic detections.

When MotoSIL does not detect any target, the *MotoSIL* item remains in the main menu bar as shown below.



After an successful target loading, the *Target* or *MotoSIL* menu item is replaced by SDI.

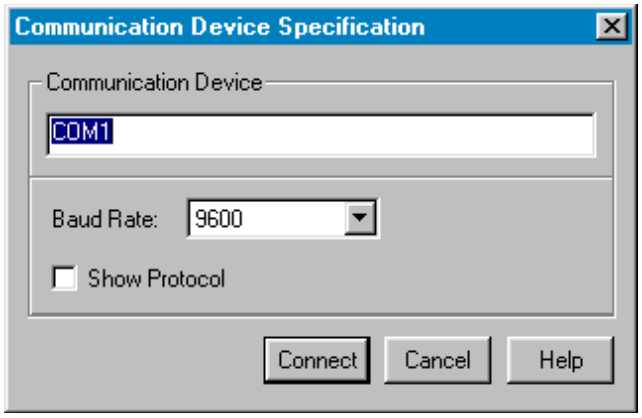


Communication Configuration

In a general way, the communication between HI-WAVE and SDI is set automatically. In case of communication setup default, a dialog pop ups and lets the user set the communication parameters.

Make sure that the parameters on your host computer are correctly configured. Check that the serial communication device used is set correctly, otherwise, any communication between HI-WAVE and the target is impossible.

Communication Device If host and target are not connected, or the connection is not via the communication device expected, a dialog box pops up in HI-WAVE as shown below.



Choose an available communication device, type it in the *Communication Device* edit box, set the Baud Rate with the drop down control and click *Connect*. This command only tries the communication device that you chose. If no connection can be established, a message box is displayed and then you can try a new communication device. Eventually the communication device chosen is saved for a later debugging session. When clicking *Cancel*, the dialog box and the environment can be quit. The default device is COM1.

Note: The communication device and the baud rate saved through this dialog override environment variables `BAUDRATE` *and* `COMDEV` *of the* `DEFAULT.ENV` *file.*

Data Format The data format used is 8 data bits, 1 stop bit, no parity and variable baud rate. The default speed is 9600 baud unless redefined using the menu entry *SDI | Communication...* in the Target menu (please see dialog below).

Communication speeds from 1200 to 57600 baud are available, depending on the hardware capabilities of host.

Default Target Setup

As any other target, the SDI target component can be loaded from the *Target* menu or can be set as a default target in the `PROJECT.INI` file which should be located in the working directory.

Example of `PROJECT.INI` file.

```
[DEFAULTS]
Window0=Source     0   0  50  40
Window1=Assembly  50   0  50  40
Window2=Register  50  40  50  30
Window3=Memory    50  70  50  30
Window4=Data       0  40  50  25
Window5=Command    0  65  50  20
Window6=Module     0  85  50  15
Target=Motosil

[Motorola ESL]
COMDEV=COM2
BAUDRATE=57600
SHOWPROT=1
```

> *Note: For further information about the* `PROJECT.INI` *file please see HI-WAVE manual.*

Motorola ESL Parameters

Under normal usage, these parameters are once interactively set at installation in the `PROJECT.INI` file and used in the future debugging sessions.

COMDEV This parameter specifies the communication device to be used between host and MMEVS. `COM1` is the default communication device for PCs and `/dev/ttya` for UNIX systems. The following list is not conclusive:

For a PC:	Any valid communication device (`COM1,COM2,etc.`).
Example:	`COMDEV=COM2`
For SUN:	Any valid communication device (`/dev/ttya, etc.`).
Example:	`comdev=/dev/ttyb`

BAUDRATE The Baud Rate of the serial communication between the target system and the host system running HI-WAVE is preset to the default of 9600 bauds and may be overwritten by one of the following baud rates:

```
1200, 2400, 4800, 9600, 19200, 28800,
38400, 57600, 115200.
```

Example: `BAUDRAUTE=19200`

SHOWPROT This variable is set to `0` or `1` (`1` to show the communication protocol, `0` to hide it) at the first communication attempt with the target. In the *Communication Device Specification* dialog, you can check the *Show Protocol* checkbox to decide if you want to display the communication protocol in a terminal window. This option is saved under the `SHOWPROT` variable.

Please see section *Communication Configuration, Communication Device Specification.*

EEPROM Programming

In order to download code or data into the onchip EEPROM, HI-WAVE has to know the location of this EEPROM. To tell HI-WAVE the location of this EEPROM define the following two environment variables:

`EEPROM_START` defines the address of the first byte of the EEPROM memory.
`EEPROM_END` defines the address of the last byte of the EEPROM memory.

Example:
```
EEPROM_START=0x0D00

EEPROM_END=0xFFF
```

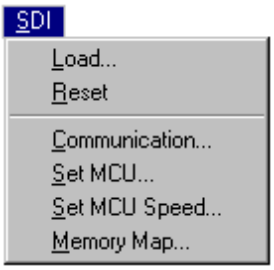
specifies the EEPROM in the memory range `0xD000` to `0x0FFF`. When writing to these addresses, the EEPROM is automatically programmed to download a program or to modify interactively the memory or the variables.

> *Note: The FLASH cannot be programmed this way!*

The HI-WAVE Status Bar for the SDI

When the SDI target component has been loaded, specific information are given in the HI-WAVE status bar. From the left to the right are displayed the baudrate of the serial communication, the HI-WAVE running mode, the E-clock frequency of the target and also the current MCU-Id.

SDI Target Component Menu Entries



Loading an application

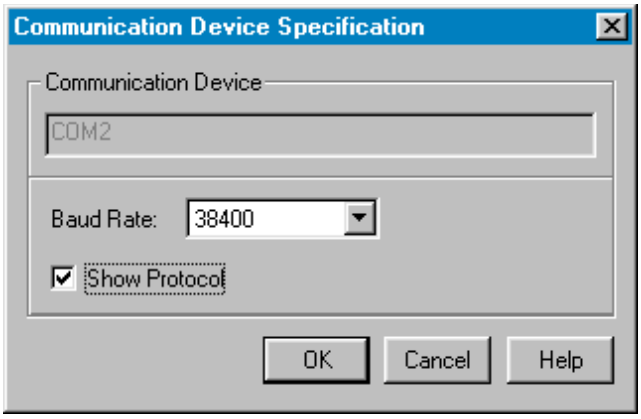
Choose *SDI* | *Load...* to load the application to debug, i.e. a .ABS file.

Communications Baud Rate

The baud rate at which the host computer communicates with the SDI should be chosen early in a session, because the system most efficiently operates when the baud rate is at the maximum supported by the host computer. This baudrate is set automatically when HI-WAVE starts to communicate with the SDI.

However, you can modify this baudrate as explained below.

Communication Select entry *SDI* | *Communication...* to display the dialog box shown below. If you know the maximum rate your host supports, select it with the drop down control (115200 is not supported on SDI). Otherwise, select 57600. If communication fails, the baud rate is automatically reduced until communication works with the host computer.



Maximum Baud Rate The maximum baud rate depends on the speed and on the

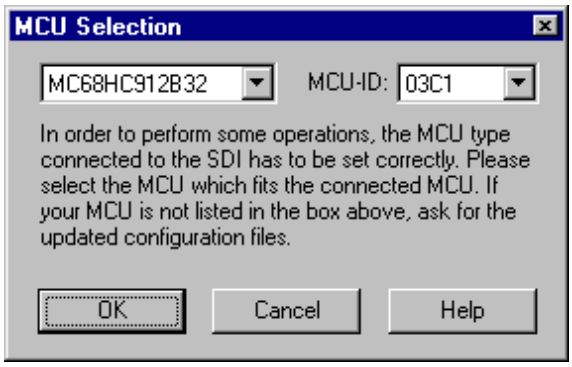
interrupt load of the host computer. On slow notebook computers or on computers running in a network the maximum baud rate may be as low as 19200. A buffered I/O card may allow the maximum rate of 57600 on any host computer. The default value is 9600.

Show Protocol If the *Show Protocol* box is checked, all the commands and responses sent and received are reported in the command line window.

> *Note: This feature is used by support personnel from Motorola or Hiware.*

MCU Selection

Choose *SDI* | *Set MCU...* to open this dialog.



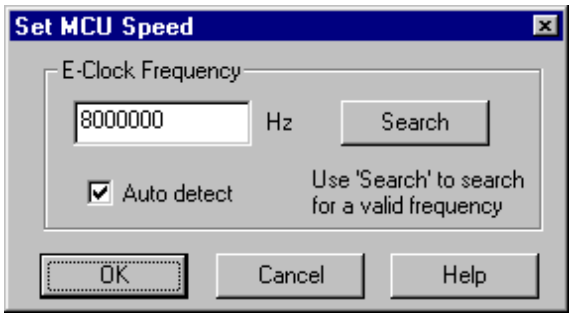
This dialog allows you to select the MCU currently used. There are two drop list Combo controls. They show the currently selected MCU name and MCU-ID.

The information will be taken from the file `MDSEMCU.INI`. If a specific MCU is not found in this file, the user is advised to update his installation.

The selection will be saved and used as default for the next session.

E-clock Frequency

Choose *SDI* | *Set MCU Speed...* to open this dialog.



This dialog show the current setting of the E-clock frequency to be used by the MCU. This frequency has to be known by the SDI for proper communication through the BDM. This is typically half of the oscillator frequency.

In the Edit box, you can specify the frequency to be used. When *Select* is clicked, the debugger tries to verify the communication and if it fails, it will search for a valid frequency in the following order:

8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz, 62.5 kHz, 6 MHz, 3 MHz, 1.5 MHz, 750 kHz, 375 kHz, 187.5 kHz, 93.75 kHz, 46.875 kHz, 7 MHz, 3.5 MHz, 1.75 MHz, 875 kHz, 437.5 kHz, 218.75 kHz, 109.375 kHz, 54.687 kHz, 5

MHz, 2.5 MHz, 1.25 MHz, 625 kHz, 312.5 kHz, 156.25 kHz, 78.125 kHz, 39.062 kHz, 32.768 kHz, 16.384 kHz.

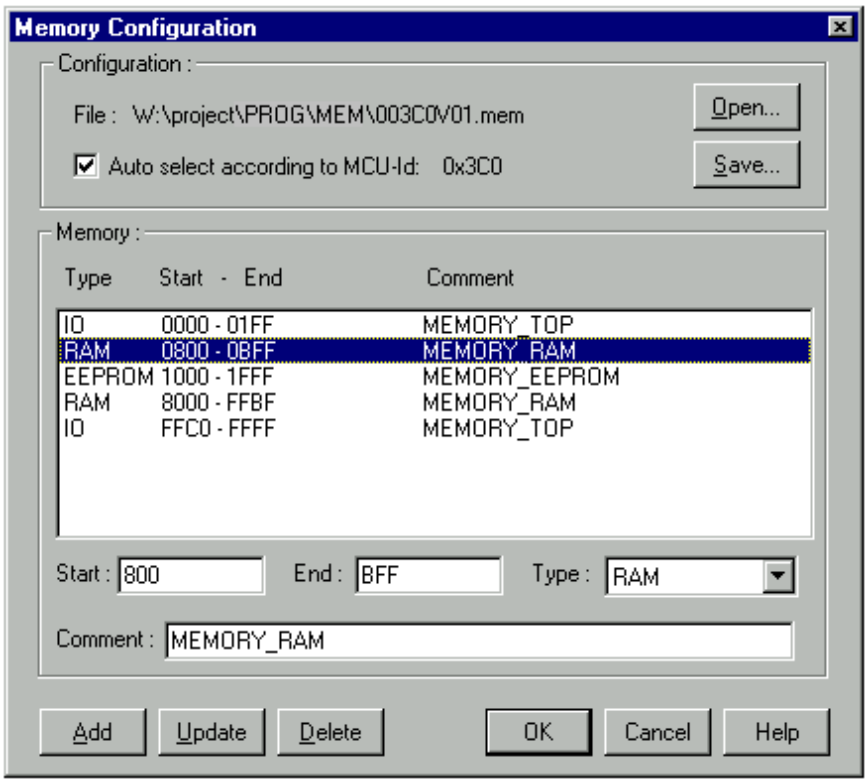
If the communication could not be verified, a message will be displayed.

At startup, the debugger will use the specified frequency. When it fails and the check box *Auto detect* is checked, it will also try to find a fitting frequency. If it was not possible to establish proper communication, an error message will be displayed, followed by this dialog.

Selections will be saved and used as default for the next session.

Memory Configuration

Choose *SDI* | *Memory Map...* to open this dialog.



This dialog shows the default configuration for the configured MCU. This setup is automatically loaded if the *Auto select* box is checked. The information about the memory layout is read from the MCU specific personality file. The personality file can be decomposed in the following way:

`00nnnVvv.MEM`

where `nnn` is the hexadecimal representation of the MCUid (3 digits) and `vv` is the version number.

This file is looked up in the "PROG\MEM" subdirectory of your installation.

> *Note: At the moment, this dialog is just used to display the default memory layout. Therefore it is not possible to modify this configuration and all buttons are disabled except "Ok", "Cancel" and "Help" buttons.*

SDI Target Startup File

The startup command file `STARTUP.CMD` is executed by HI-WAVE straight after the SDI target driver has been loaded. This file must be located in the working directory. You can use any HI-WAVE command in this file and take advantage of the wide set of commands introduced in the *HI-WAVE Userguide*.

Example of a `startup.cmd` file content:

```
wb 0x0035   0x00
wb 0x0012   0x11
```

SDI Reset Command File

The SDI reset command file `RESET.CMD` is executed when choosing *Reset* in the *`Target/SDI`* menu. This file must be located in the working directory. You can use any HI-WAVE command in this file.

> *Note: This file is NOT executed by the* `RESET` *command when entered in the Command Line component.*

Loading the SDI Component

The SDI interface is monitored by the HI-WAVE target component. The driver of the SDI is loaded when loading the SDI target component. The SDI target DLL is not included in the HI-WAVE base installation. The SDI target component is loaded through `PROJECT.INI` or through *Component* | *Set Target...*and choosing *Motosil.*

On-chip Hardware Breakpoint

An on-chip hardware breakpoint module can be used to implement breakpoints. To invoke this module to take advantage of hardware breakpoints, it is necessary to initialize the environment variable `HWBPMODULEADR` in the `DEFAULT.ENV` file

with the address in memory of the hardware breakpoint module. Make sure that the line is not in remark. If not using hardware breakpoints and the associated module, simply remove this line.

Example:

```
HWBPMODULEADR=0x20
```

This example shows the value to be used for the M68HC12B32.

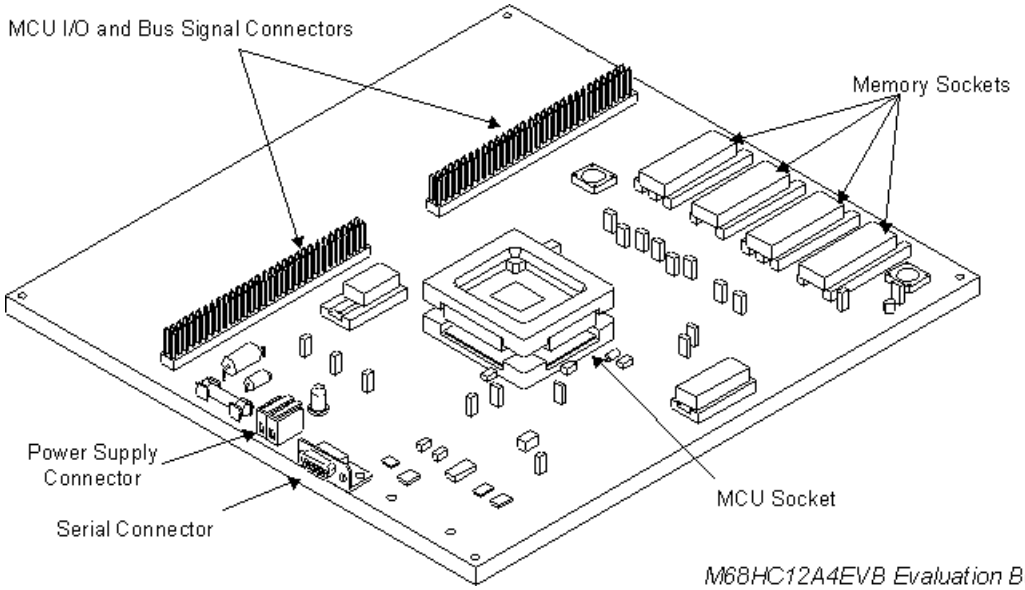
.

Note: The SDI hardware breakpoint can only handle 2 breakpoints at the same time. Additional breakpoints will be considered as software breakpoints.

Note: If you are debbuging your code in FLASH, you should not set more than 2 breakpoints.

Note: Some actions like “stepping over” or “stepping out” use one internal breakpoint and therefore reduce your amount of hardware breakpoint to 1.

Running Motorola's EVBs with the SDI



M68HC12A4EVB Evaluation Board

Introduction

The SDI can be used together with any target system which is equipped with a Background Debug connector (BDM). Motorola's Evalution Boards also called "*EVB*" are equiped with this BDM connector and therefore can be connected to the SDI interface.

Please see also section *Interfacing Your System and a Target.*

MC68HC812A4EVB Evaluation Board

This Evaluation Board for the HC12 from Motorola is prepared to work with the SDI and therefore has a BDM connector. It supports a HC12A4 processor.

- **Hardware setup** (taken from the ***HC12AEVBUM/D*** hardware manual from Motorola, ***Appendix F***) to run the SDI with the MC68HC812A4EVB:

1. Remove the jumper on header W11 from CSD.
2. Move the CSP0 jumper on W11 to pins 2-3.
3. Remove the BKGD jumper (W30).
4. The MODA jumper (W34) must shortcut the pin 1 and pin 2.
5. The MODB jumper (W42) must shortcut the pin 1 and pin 2.

- **Software setup** to be inserted in the `RESET.CMD` file of your current project directory:


```
wb 0x000B 0xF0 //MODE: forcing the Normal Exp Wide mode
wb 0x000B 0xF0 //    : this forcing must be done twice!!
wb 0x0011 0x00 //INITRG
wb 0x0010 0x08 //INITRM
wb 0x0012 0x11 //INITEE, move on-chip EEPROM from $F000
               //        to $1000
```

> *Note: You will notice that the hardware setup steps 4 and 5 (given by Motorola) force the HC12A4 operating mode to* ***Single Chip****, this to enable to start the background debugger. Then you must* ***force by software*** *(as shown above in the* `RESET.CMD` *file) the* ***Expanded Wide*** *mode to be able to access the on-board 16 Kbytes RAM mapped from* `$C000` *to* `$FFFF`.

After the setup given above, the new SDI memory map:

`$0000-$01FF`, CPU registers, on-chip (MCU)

`$0800-$0BFF`, user data area, 1 Kbyte on-chip RAM (MCU)

`$1000-$1FFF`, user code area, 4 Kbyte on-chip EEPROM (MCU)

`$C000-$FFFF`, user code/data area, 16 Kbytes external RAM (U4, U5A)

Please see also the SDI hardware manual, *SDI INTERFACE USER'S MANUAL* from Motorola and the MC68HC812A4EVB manual from Motorola.

All demo programs delivered on the SDI installation disk can be loaded and run with SDI on the EVB Board.

MC68HC912B32EVB Evaluation Board

This Evaluation Board for the HC12 from Motorola is prepared to work with the SDI and therefore has a BDM connector. It supports a HC912B32 processor.

To run the SDI with the MC68HC912B32EVB:

-You do not need to set any jumper.

-The ROM monitor program can be directly downloaded by software to the processor EEPROM if necessary.

Please see also the SDI hardware manual, *SDI INTERFACE USER'S MANUAL* and the MC68HC912B32EVB manual from Motorola.

Connection between SDI and HC12B32EVB

Some boards have a 6-pin BDM male connector and some have a 10-pin BDM male connector (HC12A4EVB). HC12B32EVB boards have a 6-pin BDM male connector. Only the 6-pin female connector of the ribbon cable (link between SDI and EVB) is used.

On the ribbon cable, the red wire is the wire number 1. Note that it is only used with the 10-pin connector. The 6-pin link/connector uses wires 4 to 9. The wire number 4 must match the pin 1 of the EVB board **BDM-in / W9** connector.

Pin 1 of the male **BDM-in** connector of EVB boards is identified by the number “1” written on the board or by a square pad (cupper side of the board) under the connector.

Power Supply

Standard Interface Hookup, Low-Voltage Interface Hookup and Alternative Hookup Conversions are precisely described in *SDI INTERFACE USER’S MANUAL*, section 2, *INTERFACE HOOKUPS.*

Appendix

SDI Commands

BAUD

Short Description

sets the communication baud rate

Syntax

BAUD [rate]

rate: Specifies the new baud rate, and must be an integer constant with one of the following specific values (decimal):

1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600.

Description

The **BAUD** command sets or displays the baud rate for communication between the system controller and the host computer. For maximum performance, the baud rate should be set as high as the host computer can accommodate. The maximum rate is 57600; the default baud rate is 9600.

If no baud rate is entered, the command displays the Communications Baud Rate Specification dialog window for interactive selection of the baud rate. If the host computer is unable to support the requested baud rate, an “Out of synchronization” alert dialog box with two options is displayed. If the ABORT option is selected, the host software exits to windows. If the RETRY option is chosen, the 9600 baud rate is used. The baud rate dialog window can also be selected from the menu.

example: BAUD 57600

Changes the communication baud rate to 57600, the maximum rate.

RESET

Short Description

resets the MCU

Syntax

RESET [**GO**|**STOP**]

where **GO** starts the target processor after reset and **STOP** does NOT start the target processor after reset.

Description

The **RESET** command resets the emulator and the emulation processor. If **STOP** is specified, the target processor is reset but stays halted. If **GO** is specified, the target processor starts running immediately after the reset and no further initialization is performed (RESET.CMD is NOT executed).

If no parameter is specified, the reset is done in the same way as any previous reset command. This is also true if you perform a reset from the target menu e.g. choosing *SDI |Reset*.

After the reset is asserted, the RESET.CMD command file is executed.

Index

T

W

