
EVB2148AF

USER MANUAL

FOR H8S/2148AF
ON-CHIP FLASH MICROCONTROLLER

PREFACE

Product Warranty

The warranty periods against defects in materials and workmanship are as set out in the accompanying Customer Information sheet.

Limitation of Warranty

The foregoing warranty does not cover damage caused by fair wear and tear, abnormal storage conditions, incorrect use, accidental misuse, abuse, neglect, corruption, misapplication, addition or modification or by the use with other hardware or software, as the case may be, with which the product is incompatible. No warranty of fitness for a particular purpose is offered. The user assumes the entire risk of using the product. Any liability of Hitachi Micro Systems Europe Limited is limited exclusively to the replacement of defective materials or workmanship.

Restrictions

Hitachi Micro Systems Europe Limited's products are not authorized for use in medical applications without prior written consent. Such use includes, but is not limited to, life support systems.

Hardware Considerations

Appropriate care should be taken not to expose the EVB to moisture, extremes of temperature or electrostatic discharge. ***Before opening the case, ensure that the unit is disconnected from the power supply to prevent accidental damage.***

Grounding

This hardware is designed for use with equipment that is fully grounded. Ensure that all equipment used is appropriately grounded. Failure to do so could lead to danger for the operator or damage to equipment.

1. Electrostatic Discharge Precautions

This hardware contains devices that are sensitive to electrostatic discharge. Ensure appropriate precautions are observed during handling and accessing connections. Failure to do so could result in damage to the equipment.

2. Electromagnetic Compatibility

Operation of this hardware with any casing removed invalidates the conformity of the equipment to the Electromagnetic Compatibility Directive 89/336/EEC.

It is advised that in this mode of operation suitable EMC precautions be observed.

Cautions

1. This document may be, wholly or partially, subject to change without notice.
2. All rights reserved. No one is permitted to reproduce or duplicate, in any form, a part or this entire document without Hitachi Micro Systems Europe Limited's written permission.

Trademarks

1. General

All brand or product names used in this manual are trademarks or registered trademarks of their respective companies or organisations.

2. Specific

Microsoft, MS and MS-DOS are registered trademarks and Windows and Windows NT are trademarks of Microsoft Corporation.

IBM is a registered trademark of International Business Machines Corporation.

Lid and Chassis Removal

The metal chassis lid can be removed by extracting the four-crosshead screws around the sides - two on each side of the lid. This gives access to all the jumpers on the EVB.

The PCB itself is held in the chassis by three crosshead screws that attach the PCB to the steel pillars. The Serial Port is held in place by two hexagonal nuts attaching it to the front of the chassis. The power supply wires from the 4mm sockets at the rear of the chassis are attached to the PCB via a connector block.

To remove the PCB completely from the chassis:

1. Remove the three screws from the PCB holding pillars.
2. Undo the connector block screws and retract the two wires **BE SURE TO MAKE A NOTE OF THE POLARITY, IT MAY NOT BE MARKED CLEARLY ON THE PCB.**
3. Extract the two hexagonal nuts from the front of the chassis either side of the connector.
4. Retract the PCB from the front of the chassis towards the back until the connector, LEDs and switches are clear.
5. The PCB can now be removed from the chassis.

Removing the FLASH Programming board

The FLASH Programming board is connected by a 14-way connector to the main PCB. To remove it, first remove the metal chassis (see above), and then pull the two PCBs gently apart.

Document Information

Product Code: D002902_11

Version: 1.0

April 2001

Copyright © Hitachi Micro Systems Incorporated. 1994-2001. All rights reserved.

Copyright © Hitachi Micro Systems Europe Ltd. 1995-2001. All rights reserved.

Copyright © Hitachi Europe Ltd. 1995-2001. All rights reserved.

Table of Contents

GETTING STARTED SECTION.....	6
INTRODUCTION	6
BUILDING AN EXECUTABLE FILE	6
DOWNLOADING AN EXECUTABLE FILE	7
TO DOWNLOAD YOUR EXECUTABLE FILE TO THE H8S/2148AF DEVICE	8
FLASH PROGRAMMING THE DEVICE	8
1 INTRODUCTION	10
1.1 EVB Functional Blocks	11
1.2 Specifications	13
1.2.1 General	13
1.2.2 RS-232 Communications	13
1.2.3 Power	13
1.2.4 PLD	13
1.2.5 Memory Map	14
2 START-UP INSTRUCTIONS	15
2.1 Installing the Low-Cost Evaluation Board (EVB)	15
2.2 Power Supply	15
2.3 Software Installation	16
2.3.1 Installation of CD-ROM Software	16
3 PRINCIPLES OF OPERATION	17
3.1 H8S/2148AF Microcontroller	17
3.2 Clock Circuitry	17
3.3 Cold Reset Circuitry	17
3.4 NMI Circuitry	18
3.5 IRQ0 Circuitry	18
3.6 SRAM	19
3.7 Serial Interface	19
3.8 LEDs	20
3.9 External User Interface	20

4	BOARD OPTIONS	26
4.1	Jumpers	26
4.2	Jumper Settings and Options	27
4.3	Setting H8S/2148AF Operating Mode (Jumpers J0 and J1)	27
4.4	Serial Port Connection (J1).....	28
4.5	FLASH Programming.....	28
4.6	Universal FLASH Programming Board Jumpers.....	29
5	CODE DEVELOPMENT	30
5.1	HOST Computer Specifications.....	30
5.2	Tutorials.....	30
5.3	Additional Information.....	31
6	HDI MONITOR	32
6.1	Introduction to HDI-M	32
6.2	Program Development.....	32
6.3	Using HDI-M.....	32
7	UNIVERSAL FLASH PROGRAMMING BOARD	33
7.1	Universal FLASH Programming Board Option	33
7.2	UFPB Interface	34
7.3	UFPB Operation	35
7.4	Transparent Mode and the PMODE pin.....	35
7.5	Operation during H8S/2148AF initialisation to BOOT, USER & Normal modes	36
7.6	Operation during Programming Kernel execution	37
7.7	Programming Software Operation.....	37
7.8	UFPB Jumper Settings and Options	37
7.9	Serial Port Connections (Jumpers JP1, JP2 and JP3).....	39
7.10	MDx Control (Jumper JP5).....	40

GETTING STARTED SECTION

PRODUCT IDENTITY: EVB2148AF

INTRODUCTION

This section is intended to help the user to get started quickly when using the H8S/2148AF EVB. It will cover how to:

- Build an executable file.
- How to download an executable file to the board.
- How to FLASH the EVB using the Universal FLASH Programming Board (UFPB).

There are five major parts to the H8S/2148AF EVB CD:

1. Hitachi Embedded Workshop (HEW) build environment.
2. Hitachi C Compiler (evaluation copy) and IAR Compiler (demo version).
3. Hitachi Debugging Interface (HDI-M).
4. FLASH Development Toolkit (FDT).
5. Tutorials and User Manuals.

The HEW Installation on the CD contains two compilers: Hitachi C Compiler and IAR Compiler, both of which are supported by the HEW build environment. The European market supports use of the IAR compiler, whilst the US market supports use of Hitachi C Compiler (evaluation copy). In order to use the Hitachi C Compiler, an initiating file must be obtained from Hitachi Semiconductor (America) Inc. Technical Support Center by sending an email request to techsupport@hsa.hitachi.com.

If you do not have the time-key file, or it is in the wrong folder, you will get the following message when you try to build your project:

"Cannot open Time Key File; Please re-install or specify "HITACHI=directory" as an environment value:".

BUILDING AN EXECUTABLE FILE

The following instructions show you how to build an example executable file using HEW with either the IAR Compiler or the Hitachi C Compiler. You will need to have installed the Hitachi Embedded Workshop, from the CD Menu Program (Menu.exe), "Install Software\Hitachi Embedded Workshop".

- Start HEW by clicking on the Start Menu, navigate to and select "Hitachi Embedded Workshop".
- At the "Welcome" dialog select "Create a new project workspace"
- Enter a project name, e.g. FLASHER
- The default directory will be "C:\hew\EVB2148\tutorials\".

-
- Select CPU Family: "H8S,H8/300"
 - Select preferred Toolchain:
 - US Customers select: "Hitachi H8S, H8/300 Standard"
 - European Customers select: "IAR ICCH8 (H8S, H8/300H)"
 - Select Project Type: "EVB2148"
 - Select OK.

The EVB2148AF Project Generator will set up the correct environment for EVB2148AF project.

- Select type of project you prefer: "Complete Projects"
- Select tutorial for project generation: "FLASHER"
- Select "Finish"
- Select "OK"

The Workspace window shows the files associated with the Flasher Tutorial Program, i.e. source files, header files etc. To view the source file in the editor window: select "B_flasher.c".

To build an executable file:

- Select "Build...Build All" from the menubar.

The output window will indicate when the build is complete and successful.

An executable file flasher.abs (from Hitachi C Compiler) or Flasher.d37 (from IAR Compiler) file will be created that will FLASH the green LED when downloaded to the H8S/2148AF device.

Note - If you encounter the following message: "Cannot open Time Key File; Please re-install or specify "HITACHI=directory" as an environment value":

Please ensure you have the Time Key File that can be obtained from Hitachi Semiconductor (America) Inc. Technical Support Centre.

DOWNLOADING AN EXECUTABLE FILE

The following procedure explains how to download the executable file you have just built to the H8S/2148AF device.

To set up a link between the HDI-Debugger and the EVB you first need to:

- Ensure mode jumpers are set for default mode, which is mode 2, the advanced expanded mode, (jumper settings for J0-J1 are found at the end of this section)
- Connect the host PC to the User (SCIO) port on the EVB and power up the EVB.
- Run HDI-M by clicking on the Launch Debugger icon on the HEW toolbar
- Create a new session on "H8S/2000 Monitor"

-
- A successful link between the HDI-M debugger and the EVB will be indicated with a “Link Up” message in the Status Bar (found in the bottom left hand corner of the HDI-M window).

Note - If the EVB cannot link up by itself a “Monitor Setup:” dialog box will appear, insert the COM port information and the baud rate – 115200 and select OK.

- If, after continued attempts the debugger fails to connect, it is probable that HDI-M monitor is not loaded into the EVB. To load the HDI-M monitor into the H8S/2148AF device please run through the section: “FLASH Programming the Device”. This will reprogram the FLASH device on the EVB2148AF with the HDI Monitor embedded kernel.

TO DOWNLOAD YOUR EXECUTABLE FILE TO THE H8S/2148AF DEVICE

Select from the HDI menu: “File...Load Program...” your preferred file from directory: “C:\hew\EVB2148\tutorials\flasher\flasher\Debug”

- US Customers select “flasher.abs”
- European Customers select “flasher.d37”
- A box will appear providing details of memory “Areas Loaded”. Select OK.

To run the program the following steps must be taken:

- Select “View...Source” from the menubar and select the source file “B_ flasher.c” from C:\hew\EVB2148\tutorials\flasher\flasher.
- Click on the “main” label of the source file and select “Run...Set PC to Cursor”. A yellow highlight will appear over the line of code.
- Select “Run...Go” from the HDI menubar.
- The green LED on the EVB will start flashing.

FLASH PROGRAMMING THE DEVICE

This process will be necessary if the user wishes to connect to the EVB with HDI monitor. Please connect the EVB device in the BOOT mode ready to reprogram the FLASH, and download the correct program as follows:

- Install the FLASH Development Software Tool (FDT) from “Install Software\Flash Development Toolkit”.
- Apply power to the EVB, this is done via the red and black socket located at the rear of the unit, (+5V only).
- Connect the serial cable to a COM port on the PC and to the SCI1 (different than the normal downloading port) connector on the EVB. The jumpers on the EVB must be in the default position (factory settings).
- Run the FDT software from the Start menu.
- Select “Open an existing image file” in order to download a single file.

-
- The lower box in the window will allow you to select "More files...". Use this to open the S-record HDI-Monitor file "2148em2.mot". This can be found by exploring the CD-ROM drive, HDIM directory.
 - To download the image select "IMAGE...Download Image..." and configure the device:
 - Target Device: H8S/2148AF
 - User Settings: COM port connected to the PC
 - Use Default Settings: UNCHECKED
 - Baud Rate: 38400 BAUD
 - Connection: BOOT
 - Select Interface: HMSE UFPB
 - Kernel already resident: UNCHECKED
 - Click the "CONNECT" button. The FLASH device is now in BOOT mode and ready to be programmed.
 - Select "Download file 2148em2.mot to device".
 - You are asked if you wish to perform a readback; select "Y"
 - The H82148 advanced expanded monitor kernel for the CPU is now programmed into the FLASH memory of the H8S/2148AF device. You can now link to HDI and download files as described above.
 - Select "Disconnect" for ending the programming procedure.
 - Close the FDT programming software and reset the EVB by pressing the CRES switch on the EVB.
 - Now switch off the EVB power supply.

Please note it is important to disconnect the FDT software from the EVB before pressing the CRES switch. Not doing so may corrupt the program and necessitate repeating the process.

1.1 EVB FUNCTIONAL BLOCKS

The EVB is composed of an H8S/2148AF single-chip microcontroller, SRAM, PLD, a FLASH programming interface board, and two serial ports. (Figure 1.2).

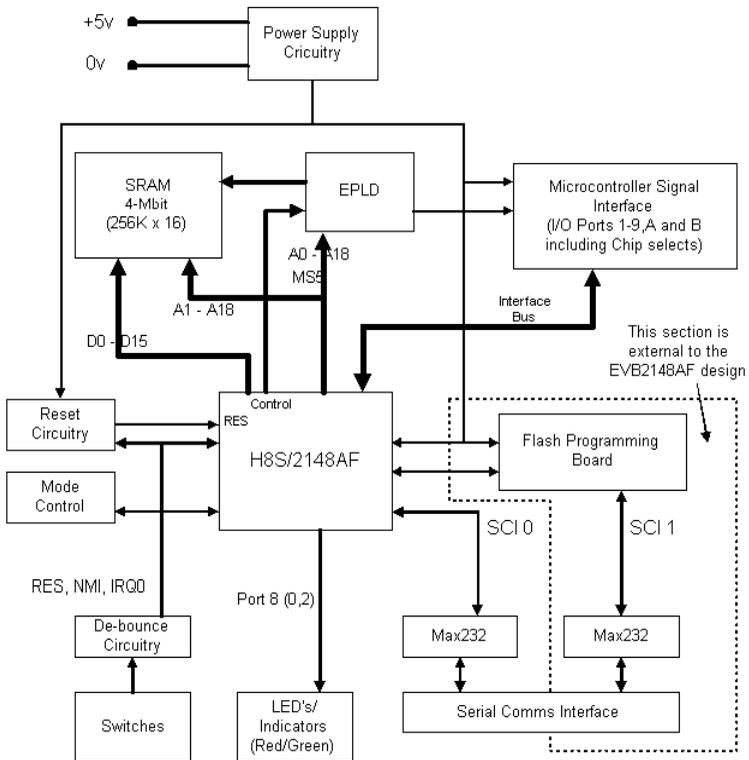


FIGURE 1.2 EVB FUNCTIONAL BLOCK DIAGRAM

The PLD contains the decoding necessary to implement an expanded memory H8S/2148AF based system. This includes the generation of chip selects and the memory control lines. The EVB is shipped with the HDI Debug Monitor firmware (HDI-M) programmed into the on-chip FLASH memory. A 4Mbit two byte-wide SRAM device (256kbyte by 16 bit) provide word-wide reads and writes. Two three-wire serial ports are supported; both serial interfaces use a serial communications channel from the H8S/2148AF device. One port (SCI1) is used to program the on-chip FLASH memory via the FLASH Programming Board, the other (SCI0) is the Host Port.

The Windows debugger HDI will connect to HDI-M to provide high level debugging via the Host port.

Users reconfiguring the EVB I/O ports are cautioned that pull-up resistors may be required for proper operation in some configurations. In addition, when connecting external analogue signals, it is important that the EVB is configured properly with respect to analogue voltage supply and reference.

1.2 SPECIFICATIONS

1.2.1 GENERAL

- 18.432MHz H8S/2148AF processor (using HD64F2148AFTE20 device)
- 512Kbytes of External SRAM, accessed in 16-bit words
- 128Kbytes of FLASH with the HDI-M firmware monitor
- 4Kbytes of internal RAM
- Two LED indicators, and three push button switches
- Detachable FLASH programming interface board
- All practical H8S/2148AF signals available for user connection

1.2.2 RS-232 COMMUNICATIONS

- Host (SCI0) interface via RS-232 DB-9S connector
- User (SCI1), FLASH Programming interface via RS-232 DB-9S connector

1.2.3 POWER

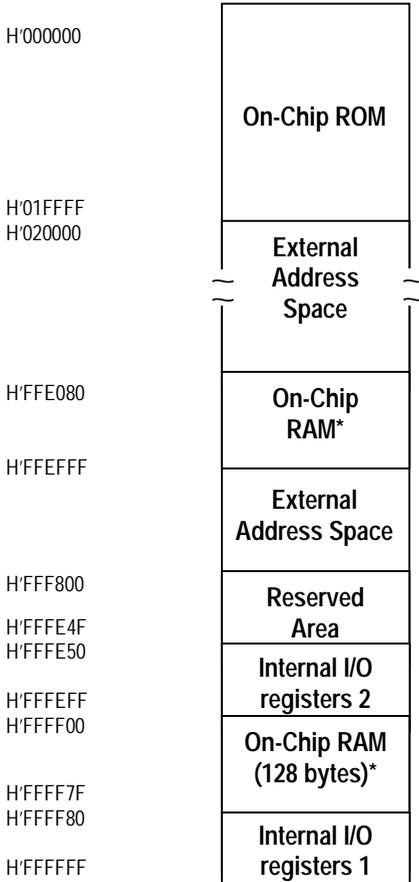
- 5V DC-only power supply
- Power connection via 4mm standard plugs

1.2.4 PLD

An Altera EPM7032STC-10 CPLD is used to perform the simple logic functions required to implement the system. The device is the ISP version of MAX 7000 and is a 44-pin TQFP package. The 'S' suffix indicates this device can be used for In-system programming. The CPLD contains logic required for minimal functions on the board i.e. switch de-bouncing, Chip select, memory control lines, address decoding, and reset control.

1.2.5 MEMORY MAP

Figure 1.3 shows the EVB memory map for the default monitor.



NOTES: *On-Chip RAM can be used as external address space by clearing the RAME bit in the SYSCR to 0.

FIGURE 1.3 EVB MEMORY MAP

2 START-UP INSTRUCTIONS

2.1 INSTALLING THE LOW-COST EVALUATION BOARD (EVB)

Installing the EVB requires power and serial connection to a host computer. The serial communications cable for connecting the EVB to a host computer is supplied. The serial connection cable has a 1:1 connectivity.

Figure 2.1 shows how to connect the EVB to a PC or notebook computer equipped with a "mini" DB-9P connector.

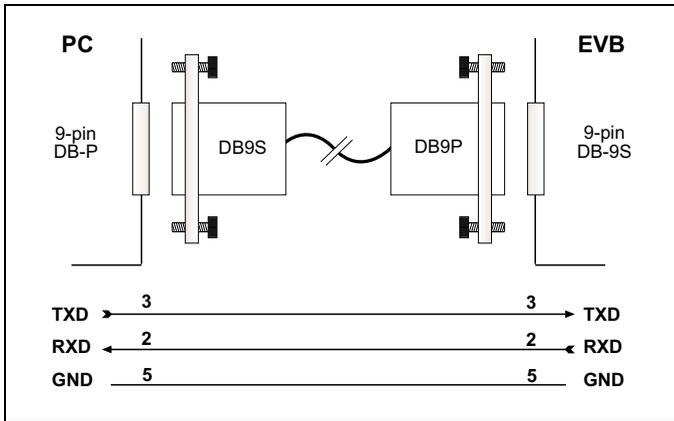


FIGURE 2.1 SERIAL CONNECTION TO PC/NOTEBOOK WITH MINI DB-9P CONNECTOR SUPPLIED

Users may also use their own serial connection cables, if desired. In the case, where a crossed cable connection is used, (i.e. TXD to RXD from Host to EVB, and TXD to RXD from EVB to Host), jumper J9 is used for Host port. The jumper is set for direct connection as default (J9: 1-3 and J9: 2-4) setting. For crossed cable connection, set jumper J9: 1-2 and J9: 3-4.

2.2 POWER SUPPLY

The EVB hardware requires a power supply of +5V supplied to CON11. Since total power consumption can vary widely due to external connections, H8S/2148AF port state, and memory configuration, the power supply must be capable of providing a maximum current of 500mA at +5V DC $\pm 5\%$.

The design includes circuitry for a Voltage supervisor to ensure that supply levels do not drop below levels that would affect MCU operation. In addition, circuitry is provided for supply filtering/noise reduction protection and reversed polarity

protection. The power connector on the EVB is connected to 4mm sockets mounted in the casing. Standard 4mm plugs may be used to connect to a bench supply.

2.3 SOFTWARE INSTALLATION

Please review the following procedures thoroughly before installing. Software is supplied with the EVB on a CD-ROM. Users should make sure their system has a CD-ROM drive correctly installed before continuing.

The CD-ROM includes the following software:

1. HDI-M Windows Debugger
2. Hitachi Embedded Workshop including Hitachi/MCS and IAR Series C compiler
3. FLASH Development Toolkit (FDT)
4. EVB Tutorials

2.3.1 INSTALLATION OF CD-ROM SOFTWARE

An installation utility is supplied with the CD-ROM, enabling the user to easily install all components of the EVB2148AF. For using this installation utility, please refer to the CD-ROM insert.

3 **PRINCIPLES OF OPERATION**

The EVB is composed of the following components:

- H8S/2148AF microcontroller
- Clock circuitry
- Cold Reset (CRES) Circuitry
- NMI circuitry
- IRQ0 circuitry
- PLD
- On-board SRAM
- Detachable Programming Interface board
- 2 x Serial Interfaces (SC10 and SC11)
- LED driver
- External User Interface

The complete EVB schematics are provided as part of the kit and are referenced throughout this chapter.

3.1 **H8S/2148AF MICROCONTROLLER**

The H8S/2148AF provides on-chip many of the functions required to implement an expanded memory microcontroller system. The address area decoding is performed by a PLD. Users reconfiguring processor I/O ports are cautioned that pull-up resistors may be required for proper operation in some configurations. In particular, users adding external memory should be aware that the chip selects provided by the PLD are shared and may be floating until the system is configured.

3.2 **CLOCK CIRCUITRY**

The clock circuitry comprises the H8S/2148AF oscillator and a 18.432MHz AT-cut parallel resonating crystal. The system clock (CK out pin) output frequency is the same as that of the internal clock.

3.3 **COLD RESET CIRCUITRY**

The reset generator for the EVB is based on the 7705A Supply Voltage Supervisor from Texas Instruments or SGS Thomson. This chip is specifically designed for use as a reset controller in microcomputer and microprocessor systems.

During power-up the device monitors the supply voltage and keeps the RESET and RESETn outputs active (high and low respectively) as long as the supply voltage has not reached its nominal value. After the voltage has reached the tolerance, the RESET and RESETn are kept active for an additional 60ms to allow for final supply stabilisation and processor reset. Typically, the H8S/2148AF needs minimum of 20ms for the crystal oscillator to stabilise.

The threshold voltage is 4.55V. Upon detection of an under voltage condition the device will assert reset to the MCU, until the levels are returned to a valid voltage. The device also has a delay, controlled by a timing capacitor, which results in a delay of approximately 60mS. There is also a push-button switch (SW1) to generate a manual cold reset (CRESn). The switch is debounced within the PLD and fed to the RST_IN input of the 7705A. When the push button is released the 7705A will provide a 60ms reset signal.

The de-bounced CRES output is ANDed with the FLASH_RES signal from the FLASH programming board to generate the final RESETn signal to the CPU. This RESETn signal is also fed back to the FLASH programming board. If the programming board is not present there is pull-up resistor to deactivate the FLASH_RES signal.

In loader mode the CRES will reset the entire board and start the bootloader program running. Therefore a new s-record file must be downloaded to the H8S/2148AF.

Quickly switching power "off" and then "on" again to the board, may not allow Vcc to fall low enough to generate a reset pulse. In practice, the H8S/2148AF usually continues to operate normally. **Rapid on/off switching of the power supply stresses the integrated circuit components and is not recommended.**

3.4 NMI CIRCUITRY

The NMI input of the H8S/2148AF is an independent edge-triggered input. NMI may be generated on the positive or negative-going transition, depending on the setting of the System Control Recognition (SYSCR) NMIEG bit. Default after reset is negative going edge.

On the EVB there is a push-button switch SW2 that is debounced by NAND gates, to generate the NMI signal. The quiescent state is low and when SW2 is pressed the NMI goes high. When SW2 is released, the NMI will return to a low level and generate a negative going edge.

The NMI signal can alternatively be driven from an external input applied to CON3 pin 9 and Jumper J7.

3.5 IRQ0 CIRCUITRY

Setting bit 0 of the IRQ Enable Register (IER) to 1 will enable IRQ0 and ensure that an interrupt will be generated when IRQ0 is activated.

IRQ0 can be set to generate an interrupt request either by low level sensing, or by falling edge, rising edge or both edge sensing. Low level, Falling edge, Rising edge, or both edge selection is possible by setting the appropriate bits (Bit 0 for IRQ0), in the IRQ Sense Control Registers A = ISCRL and B = ISCRL, (in both registers, IRQ0 is set by Bit 0 of both Sense control A & B). The default value is to generate an interrupt request on an input low level.

On the EVB there is a push-button switch, SW3 that is debounced with NAND gates, with Schmitt trigger input and the output signal is connected to the IRQ0 pin of the H8S2148AF. An external IRQ0 signal can also be applied to CON3 pin 13 of if the Jumper J8 is in between position 2 & 3 The quiescent state is high, and when SW3 is pressed the IRQ0 will return to a low level and generate a negative going edge.

When the board is in Run mode, in which the downloaded program is being executed, the IRQ0 switch (SW3) will cause the H8S/2148AF to interrupt the operation of the program.

3.6 SRAM

The SRAM configuration is 256k x 16 bits, external logic is required on the board to provide the required signals for interfacing to 16bit wide SRAM. This is done by the PLD, described in section 5. (Figure 1.3 Section 1 shows the memory map).

The SRAM is only accessible in the advanced expanded Mode 2 (in CS0 space). The Memory is broken up into seven addressable areas, each 2Mbytes in size (with the exceptions of CS0 which is 524Kbyte and CS6 which is 1Mbyte) and accessed using chip selection. The SRAM is located at H'200000 to H'27FFFF and the memory available for the user is 256Kbytes. When RAM is referenced at this address, the memory area chip select signal CS0n, is generated by the PLD

3.7 SERIAL INTERFACE

The EVB supports two three-wire serial channels using the two serial communications channel (SCI0 and SCI1) in the H8S/2148AF CPU. SCI0 is dedicated to use by the HDI-M for communications with a host PC. While SCI1 is normally the USER port and used for Flash Programming.

SCI0 from the H8S/2148AF is connected via an RS232 transceiver to a 9-way D-type connector, CON1 on the main board. Transmit and Receive signals from SCI0 are connected via the RS232 transceiver and dedicated to the main host PC. The MAX232 transceiver device translates RS232 signals to logic levels and vice-versa.

SCI1 from the H8S/2148AF is connected via an RS232 transceiver on the FLASH programming board, to a 9-pin filtered D-type connector also on the FLASH programming board. On this board a transceiver (MAX232) circuit translates to RS-232 levels.

TABLE 3.1 RS-232 INTERFACE SIGNALS

Connector	Signals
CON1	SCI0
J5 (UFPB)	SCI1

3.8 LEDs

There are two LED's on the EVB to indicate board status. The first LED is green and can be driven, using port 8 bit 0 of the H8S/2148AF. The second LED is red, and indicates when power is supplied to the EVB. The user can select to drive this LED from the MCU output port 8, bit 2 when jumper J8 is in between 2 & 3.

3.9 EXTERNAL USER INTERFACE

The external user interface makes most of the H8S/2148AF signals available to the user consistent with keeping:

- Signal lines short
- Board design simple
- Functional signals grouped together
- Lines potentially used for analogue signals isolated

The external user interface connectors are 20 pin 0.1" pitch headers allowing user interface to the respective ports of the H8S2148AF, plus some control signals. Commonly available 2.54mm (0.100 inch) male headers with 0.635mm (0.25 inch) square posts are used. Tables 3.3 to 3.9 define the signals available on each of the connectors 2 – 8 respectively.

TABLE 3.2 INTERFACE CONNECTORS, ANALOGUE, SERIAL AND POWER SIGNALS

Connector	Signals
CON1	9-way D-type (RS232)
CON9	14 Way Female header connector for connecting to FLASH programming board.
CON10	Altera ISP Connector
CON11	Power Connector

Note: Each of these external interface connectors also includes VCC, normally at +5V and GND. Trivial external circuits may use the power from the EVB. However, if more than 50mA is needed the external circuits should be powered by an independent power supply.

Figure 3.1 shows the board locations of these connectors CON2 to CON8. Note the positioning of pin 1 on each connector. The pins are numbered odd-even as shown in Figure 3.2.

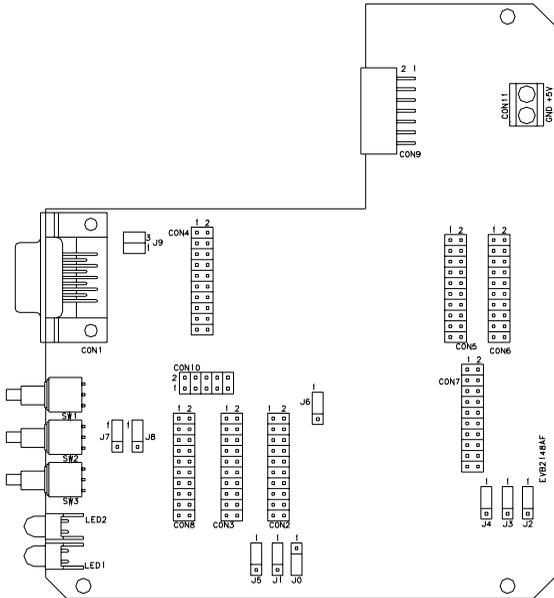


FIGURE 3.1 CONNECTOR LOCATIONS (COMPONENT SIDE)

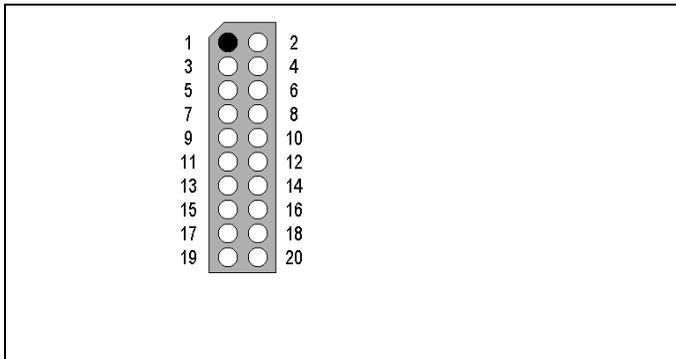


FIGURE 3.2 CONNECTOR CONFIGURATION

TABLE 3.3: PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON2

Connector Pin	Signal	Connector Pin	Signal
1	PORT1(0)	2	PORT2(0)
3	PORT1(1)	4	PORT2(1)
5	PORT1(2)	6	PORT2(2)
7	PORT1(3)	8	PORT2(3)
9	PORT1(4)	10	PORT2(4)
11	PORT1(5)	12	PORT2(5)
13	PORT1(6)	14	PORT2(6)
15	PORT1(7)	16	PORT2(7)
17	VCC	18	GND
19	VCC	20	GND

TABLE 3.4: PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON3

Connector Pin	Signal	Connector Pin	Signal
1	PORT5(0)	2	PORT8(0)
3	PORT5(1)	4	PORT8(1)
5	PORT5(2)	6	PORT8(2)
7	STBYn	8	PORT8(3)
9	EXT_NMI _n	10	PORT8(4)
11	RES0 _n	12	PORT8(5)
13	EXT_IRQ0 _n	14	PORT8(6)
15	EXT_RESET _n	16	GND
17	VCC	18	GND
19	VCC	20	GND

TABLE 3.5: PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON4

Connector Pin	Signal	Connector Pin	Signal
1	PORTA(0)	2	GND
3	PORTA(1)	4	CS(0)
5	PORTA(2)	6	CS(1)
7	PORTA(3)	8	CS(2)
9	PORTA(4)	10	CS(3)
11	PORTA(5)	12	CS(4)
13	PORTA(6)	14	CS(5)
15	PORTA(7)	16	CS(6)
17	VCC	18	GND
19	VCC	20	GND

TABLE 3.6: PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON5

Connector Pin	Signal	Connector Pin	Signal
1	PORTB(0)	2	PORT3(0)
3	PORTB(1)	4	PORT3(1)
5	PORTB(2)	6	PORT3(2)
7	PORTB(3)	8	PORT3(3)
9	PORTB(4)	10	PORT3(4)
11	PORTB(5)	12	PORT3(5)
13	PORTB(6)	14	PORT3(6)
15	PORTB(7)	16	PORT3(7)
17	VCC	17	GND
19	VCC	18	GND

TABLE 3.7: PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON6

Connector Pin	Signal	Connector Pin	Signal
1	PORT4(0)	2	PORT6(0)
3	PORT4(1)	4	PORT6(1)
5	PORT4(2)	6	PORT6(2)
7	PORT4(3)	8	PORT6(3)
9	PORT4(4)	10	PORT6(4)
11	PORT4(5)	12	PORT6(5)
13	PORT4(6)	14	PORT6(6)
15	PORT4(7)	16	PORT6(7)
17	VCC	18	GND
19	VCC	20	GND

TABLE 3.8 PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON7

Connector Pin	Signal	Connector Pin	Signal
1	PORT7(0)	2	AVSS
3	PORT7(1)	4	AVSS
5	PORT7(2)	6	AVSS
7	PORT7(3)	8	AVSS
9	PORT7(4)	10	AVSS
11	PORT7(5)	12	AVSS
13	PORT7(6)	14	AVSS
15	PORT7(7)	16	AVSS
17	VCC	18	GND
19	VCC	20	GND

TABLE 3.9 PIN ASSIGNMENT FOR INTERFACE CONNECTOR CON8

Connector Pin	Signal	Connector Pin	Signal
1	PORT9(0)	2	GND
3	PORT9(1)	4	GND
5	PORT9(2)	6	GND
7	PORT9(3)	8	GND
9	PORT9(4)	10	GND
11	PORT9(5)	12	GND
13	PORT9(6)	14	GND
15	PORT9(7)	16	GND
17	VCC	18	GND
19	VCC	20	GND

4 BOARD OPTIONS

The EVB provides a number of user-definable optional configurations. Jumper settings or configuration via zero Ohm resistors chooses some of these, while others use alternative loading options.

4.1 JUMPERS

EVB jumpers allow users to configure the board as required for testing or evaluation. The jumpers come in three types: 2 or 3 pin in-line for a single jumper or 2 by 2-pin for 1 or 2 jumper capacity. The three-pin in-line jumpers are used to connect one of two possible signals to a third line, the 4-pin blocks typically allow signal paths to be changed by rotating the jumpers through 90°.

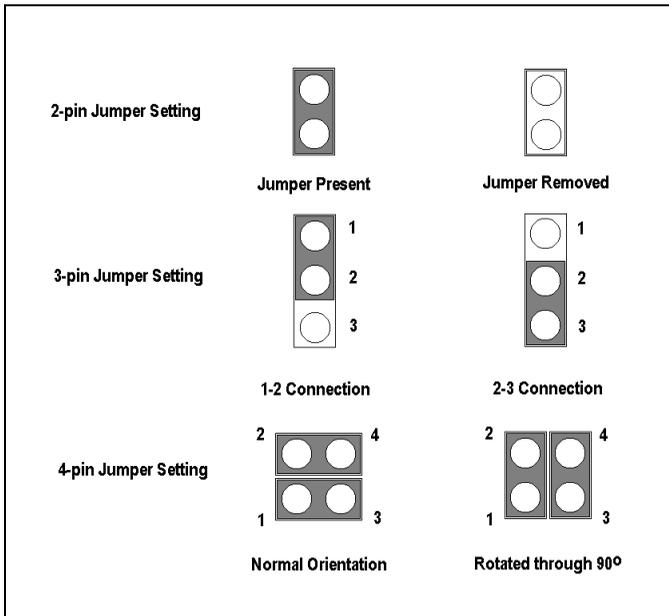


TABLE 4.1 JUMPER SETTINGS

4.2 JUMPER SETTINGS AND OPTIONS

Table 4.2 below shows the different jumpers settings for the different jumpers.

TABLE 4.2 JUMPER SETTINGS AND OPTIONS

Three-pin in-line jumpers:

Jumper	Use	Pins (1-2)	Pins (2-3)
J0	MD0	MD0 pulled to Vcc	MD0 tied to GND (default)
J1	MD1	MD1 pulled to Vcc (default)	MD1 tied to GND
J2	Avref	Avref tied to 5V (default)	External Avref selection
J3	Avref	Avcc tied to 5V (default)	External Avcc selection
J4	Avss	Avss tied to ground (Digital) (default)	External Avss selection
J5	Red LED	LED to Vcc (default)	LED to Port 8 bit 2
J6	RSTn	O/P of Voltage supervisor connected to RSTn (default)	External Reset connected to RSTn
J7	NMI	NMI Switch connected to CPU NMI Pin (default)	External NMI connected to CPUs NMI Pin
J8	IRQ0	IRQ0 Switch to CPU IRQ0 Pin (default)	External IRQ0 connected to CPUs IRQ0 Pin

Four-pin jumpers:

Jumper	Use	Default (Pins 1-3 2-4)	Alternative (Pins 1-2 3-4)
J9	TxD & Rx D connection orientation	1:1 connection	Crossed connection

4.3 SETTING H8S/2148AF OPERATING MODE (JUMPERS J0 AND J1)

As described in Section 3 of the H8S/2148AF Hardware Manual, the operating modes of the H8S/2148AF are set during device initialization, by setting the two mode pins MD0 and MD1.

The supported operating modes of the H8S/2148AF are modes 2 [Advanced (Expanded and Single-Chip)] and 3 [Normal (Expanded and Single-Chip)]. The default-operating mode of the H8S/2148AF is mode 2. (Advanced expanded mode).

Table 4.3 MCU Operating Modes

MCU Operating Mode	MD1 (J10)	MD0 (J11)	Description
*2	0	1	Advanced Expanded mode with on-chip ROM enabled (16Mbyte)
2	0	1	Advanced single-chip mode with on-chip ROM enabled (16Mbyte) NOTE- Difference between Modes is the setting of the EXPE bit in the Mode Control Register (MDCR)
3	1	1	Normal expanded mode with on-chip ROM enabled (64Kbyte)
3	1	1	Normal single mode with on-chip ROM enabled (64Kbyte) NOTE- Difference between Modes is the setting of the EXPE bit in the Mode Control Register (MDCR)

** The default-selected mode will be over-ridden automatically by the programming board*

4.4 SERIAL PORT CONNECTION (J1)

SCI0 is dedicated by default to the monitor for the HOST port, and SCI1 is dedicated to the USER port via the programming board.

SCI1 is used to communicate with the UFPB during on-board programming modes (BOOT and USER Mode). Table 4.3 defines settings for on-board programming modes of operation.

Transmit and Receive signals from SCI0 and SCI1 are connected to serial transceiver circuits.

4.5 FLASH PROGRAMMING

For H8S/2148AF FLASH programming, there are two modes of programming options:

The BOOT mode is selected the first time (an empty) H8S/2148AF is programmed and is performed before shipment of the EVB.

The USER program mode is performed when a user program in FLASH is to be updated.

4.6 UNIVERSAL FLASH PROGRAMMING BOARD JUMPERS

The UFPB is provided with a number of user configurable jumpers. These allow the users to configure as required (refer to Section 7).

Table 4.4 describes the Jumper Settings of the UFPB.

TABLE 4.4 JUMPER SETTINGS AND OPTIONS

Jumper	Use	Default Setting (1-3, 2-4)	Alternate Setting (1-2, 3-4)
JP1	TxD and RxD connection orientation	1:1 Connection	Crossed Connection

Jumper	Use	Default Setting (1-2, 3-4)	Alternate Setting (1-3, 2-4)
JP2 & JP3	Programming Board Bypass Connection	Non-bypassed connection	Bypassed connection

Jumper	Use	Setting (1-2)	Setting (2-3)
JP4	FWE/FWP Control	FWP Signal	FWE Signal
JP5	MDx Control	12V/5V	12V/0V

Note JP4 allows the user to configure the UFPB to produce either a FWE signal or a FWP signal for FLASH programming **this is not used with the H8S/2148AF**.

If the user wishes to use their own serial connection cable, in the case where a crossed cable is used (i.e. TxD to RxD from host to EVB and TxD to RxD from EVB to host) jumper JP1 provides the user with the capability to cross the RxD and TxD connections. See Table 4.4 for further details.

JP2 and JP3 allow SCI1 of the H8S/2148AF to be connected in one of three ways:

- Connected directly to the RS-232 transceiver (U4) on the UFPB (default setting)
- Connected to the SCI interface of the UFPB
- Disconnected from the UFPB

Table 4.4 details the jumper settings for the programming board connections (if the jumpers are removed, then the SCI1 port is disconnected from the UFPB).

The default option for JP5 is to have the jumper socket connected between pins 1-2. This allows the MDx signal to be either 12V or 5V depending on the mode of operation. In the alternative setting when the jumper socket is connected across pins 2-3, the MDx signal can either be 12V or 0V depending on the mode of operation. This Jumper can be left in the default position, as this pin is not used by the H8S/2148AF.

5 CODE DEVELOPMENT

The H8S/2148AF EVB is supplied with a limited evaluation version of the Hitachi Embedded Workshop (HEW) for H8S (the standard development tools for the EVB). This includes limited evaluation versions of the Hitachi/MCS & IAR H8 Series C Compiler toolchains. The Hitachi/MCS Compiler is time-limited for 30 days. The IAR Compiler is code-size limited (a maximum of 2K of code can be generated by the compiler, a maximum of 1K of code can be generated by the assembler and the final linked application cannot exceed 4K). Hitachi also makes full versions of the HEW tools available for H8S development (an extra-cost option).

When debugging programs developed for use with the EVB, use the HDI-M debugger – stored in FLASH – with the HDI Windows GUI. HDI supports either the Hitachi/MCS tools or the IAR tools.

5.1 **HOST COMPUTER SPECIFICATIONS**

For program development with the EVB package, you will need:

- 80Mb or more of hard disk space.
- A computer capable of hosting the Windows tools, editing files, and communicating with the evaluation board. The computer must be an i486, or better, standard PC running Windows 95/98/NT (Pentium recommended).
- 32Mb or more of memory.
- A CD-ROM drive. The tools and tutorials are distributed on a single CD-ROM.
- A standard serial port, COM1 or COM2 is required for host computer communication with the EVB2148AF.

5.2 **TUTORIALS**

The separate tutorial manual has more information on installing and using the Hitachi Windows tools to do simple program development and demonstrations on the EVB. Subjects including preparing and running:

- A program that turns the green LED on and off
- A program that Flashes the green LED

The tutorial manual includes information on developing code that makes use of the on-chip peripherals of the H8S/2148AF, and provides examples of initialisation of static data. These examples provide more in-depth information on the use of compiler extensions, interfacing to assembler and use of linkage control files. Subjects include:

- A simple program with startup code for initialisation of static data.
- A program to FLASH the green LED using a count loop within the program.
- A program that uses timer interrupts to vary the Flashing of the green LED.
- A program that may be programmed into FLASH, to replace HDI-M, highlighting the steps required to create real application code.

The tutorial manual details the use of the on-chip FLASH of the H8S/2148AF. The examples enable users to program the on-chip FLASH using the programming board and FLASH Development Toolkit software.

The tutorial section includes:

- Overview of FZTAT operation
- Programming board and interface software usage
- Programming tutorials from the Tutorial section into the on-chip FLASH
- Understanding the operation of the BOOT and USER mode kernels

5.3 ADDITIONAL INFORMATION

For details on how to use HDI-M, the EVBs resident debugger, refer to the HDI-M manual supplied on the EVB CD-ROM.

For information about the H8S/2148AF series microcontrollers, refer to the *H8S/2148AF Series Hardware Manual* (Hitachi order number: ADE-602-125A).

For more information about the H8S/2148AF assembly language, refer to the *H8/2600 Series, H8/2000 Series Programming Manual* (Hitachi order number: ADE-602-083).

This document is not intended to be a tutorial on embedded system programming in general, C language, or the IAR tools.

The standard reference book on the C language is *The C Programming Language*, Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1978.

6 **HDI MONITOR**

6.1 **INTRODUCTION TO HDI-M**

HDI-M is a FLASH-resident debugging monitor program hosted on the EVB. HDI-M may be used to download, run, and debug programs developed on a PC. HDI-M provides all the necessary control and communications to operate under the HDI GUI. This allows users to perform high-level C debugging on the EVB. Using HDI's powerful debugging features, users may explore features of the H8S/2148AF processor and the EVB by directly entering and running simple programs.

Install HDI-M from the CD-ROM, as described in Section 2.

6.2 **PROGRAM DEVELOPMENT**

The EVB Tutorials contain examples you may use as a basis to explore and evaluate the architecture of the H8S/2148AF. When you install the Hitachi Embedded Workshop (HEW), an EVB2148AF specific project generator wizard is installed. To generate the EVB tutorials:

- Open a new workshop.
- Select your preferred toolchain and project type – EVB 2148.

You can then choose to generate a blank project for the EVB or one of the tutorial projects, either completed or a skeleton project for completion following the tutorial manual.

6.3 **USING HDI-M**

HDI-M is a free monitor for use with the evaluation boards. A manual is supplied in PDF format on the CD-ROM covering installation and basic usage. The tutorials, in the separate tutorials manual, are specifically designed to cover embedded code development, and are not intended as a tutorial on using HDI-M. **Please refer to the HDI-M manual for further information.**

7 UNIVERSAL FLASH PROGRAMMING BOARD

7.1 UNIVERSAL FLASH PROGRAMMING BOARD OPTION

The Universal FLASH Programming Board (UFPB) is a plug-in module, which is capable of providing the control signals and voltages required to place an H8S/2148AF into BOOT, USER or normal execution mode without requiring the user to touch the EVB.

The programming board has been designed to allow users to test the features of the on-chip FLASH, without having to worry about any hardware requirements. In addition it may be unplugged from the EVB2148AF main board, and connected to a user's system using a 14-pin interface (specified in table 7.2). This allows users to evaluate In-System-Programming (ISP) of the H8S/2148AF without having to construct additional circuitry. Figure 7.1 shows the physical layout of the UFPB.

Note: Using the programming board outside of the enclosure supplied invalidates the conformity to the EMC directive (89/336/EEC). It is advised that in this mode of operation, suitable EMC precautions are taken.

Note: The programming board as supplied with the EVB kit, is for prototype use only. It should not be used in mass production to program devices.

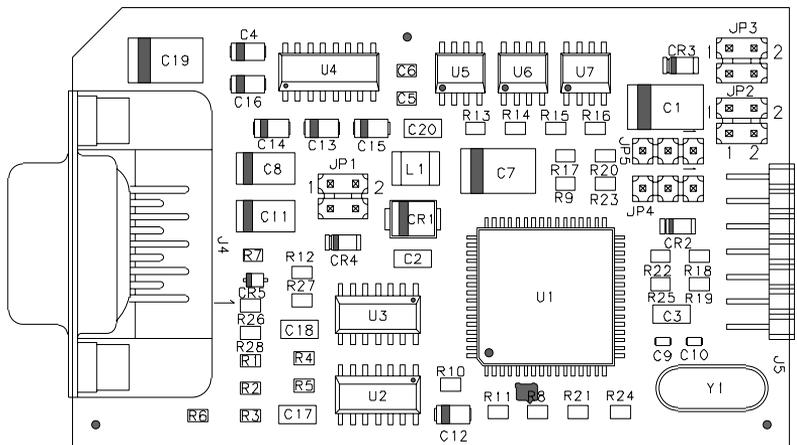


FIGURE 7.1 PHYSICAL LAYOUT OF THE UFPB

7.2 UFPB INTERFACE

The interface between the UFPB and the main board of the EVB is achieved as follows:

The male (2x7 Way) connector J5 on the UFPB connects to the female (2x7 Way) connector CON9 on the main board of the EVB. The programming board is detachable and may be used in conjunction with other Hitachi EVB's with the exception of the EVB7050 or can be used to program a H8S/2148AF device on the users hardware. If the UFPB is to be used on the users own hardware the user requires a corresponding CON9 connector with the same signal connections of the CON9 fitted on the EVB.

TABLE 7.1 PIN DESCRIPTION OF THE UFPB INTERFACE CONNECTOR J5

J5 Pin No	Signal Name	Signal Description
1	VCC	+5V supply input to UFPB
2	GND	Ground reference input to UFPB
3	RX	Output data to main board of EVB
4	TX	Input data from main board of EVB
5	WRES*	Not used
6	CRES*	Cold reset input from main board of EVB
7	Vpp	Not used
8	MDx	Not used
9	PMODE	Programming mode input from main board of EVB
10	RESET*	Reset output to main board of the EVB
11	FEW	Not used
12	Mdy	H8S/2148AF MD1 pin
13	GND	Ground connection
14	GND	Ground connection

TABLE 7.2 PIN DESCRIPTION OF THE EVB MAIN BOARD INTERFACE CONNECTOR CON9

CON9	Name	Function
1	VCC	+5V supply connection to UFPB
2	GND	Ground Connection to UFPB
3	RX	Input data from UFPB
4	TX	Output data to UFPB
5	N/C	Not used
6	RSTn	Cold Reset output to UFPB
7	N/C	Not used
8	N/C	Not used
9	PMODE	H8S/2148AF Port 8, bit 1
10	FLASH_Reset	Reset (active low) to H8S/2148AF
11	N/C	Not used
12	Mdy	H8S/2148AF MD1 pin
13	GND	Ground connection
14	GND	Ground connection

Figure 7.2 below shows the placement and pin numbering system for the CON9 connector on the EVB.

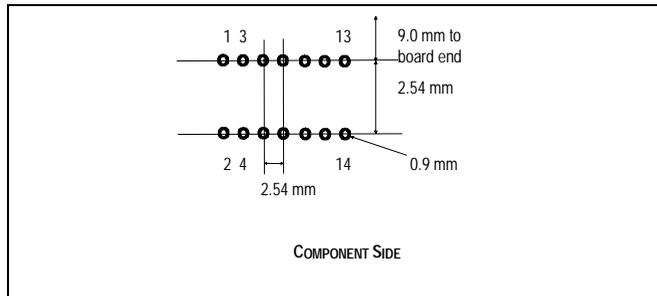


FIGURE 7.2 PLACEMENT AND PIN-NUMBERING OF THE CON9 CONNECTOR ON THE EVB

The CON9 connector used on the EVB is a double row, right angle receptacle from Samtec.

7.3 UFPB OPERATION

The H8/3217 on the UFPB board allows the Windows FLASH programming interface to dynamically control all aspects of the boards operation. The RS-232 connection is used to communicate between the H8/3217 and the host PC. The H8/3217 also connects to the SCI1 of the H8S/2148AF during programming for data and command transfer.

Please note: This type of board is not required for an end-user system, it has been designed by Hitachi to be as flexible as possible. The application note supplied details how to implement FLASH programming within the user's system.

Note: The programming board that you received in your kit can be modified to program other Hitachi FLASH devices.

As detailed in section 7.8 the H8/3217 may be bypassed using jumpers JP2 and JP3 to allow the H8S/2148AF to directly access the RS-232 port on the programming board when the user does not wish to program the FLASH memory on the device.

7.4 TRANSPARENT MODE AND THE PMODE PIN

The final operation performed during initialisation is to place the programming board into *transparent mode*. In this mode any character received from the Host is re-transmitted directly to the H8S/2148AF and vice-versa *with no processing*. The

H8/3217 effectively stops command processing, and will not resume unless reset, or signalled by the H8S/2148AF via the PMODE pin (Port 8, bit 1).

During normal command processing, the PMODE signal is ignored. The programming kernel that is downloaded to the H8S/2148AF takes this pin LOW during initialisation. During transparent mode, the H8/3217 checks the status of the PMODE signal, if this is taken HIGH, the H8/3217 leaves transparent mode. This operation is performed by the programming kernel during disconnect.

If users wish to implement their own programming kernel, then the PMODE signal (Port 8, bit 1) should be controlled in a similar manner.

When connecting the programming board to user's hardware, the PMODE signal should be connected to some form of switchable circuit, which may be controlled from software or hardware, in the way described above.

7.5 OPERATION DURING H8S/2148AF INITIALISATION TO BOOT, USER & NORMAL MODES

On power-up the programming board places all control signals into normal operation settings (MDy = 5V, RESET is held high), and the H8/3217 enters a command processing loop to wait for command packets from the FLASH programming utility. When the user selects to BOOT the H8S/2148AF, the Windows interface sends a sequence of commands to the programming board to perform the following tasks:

- Hold the H8S/2148AF RESET line low
- Connect 0V to MD1 of the H8S/2148AF
- Take the H8S/2148AF RESET line high
(The H8S/2148AF will now be in BOOT mode)
- Negotiate serial communications with the H8S/2148AF
- Download BOOT programming kernel to the H8S/2148AF
- Enter transparent mode

When the user selects to place the H8S/2148AF into USER mode, the Windows interface assumes that code is resident on the H8S/2148AF, which mimics the operation of the device in BOOT mode (without first erasing the on-chip FLASH). The initialisation sequence is then:

- Negotiate serial communications with the H8S/2148AF
- Download BOOT programming kernel to the H8S/2148AF
- Enter transparent mode

For Normal Mode execution, usually after exit from transparent mode:

- Hold the H8S/2148AF RESET line low
- Disconnect 0V from MD1 of the H8S/2148AF
- Take H8S/2148AF RESET line high

7.6 OPERATION DURING PROGRAMMING KERNEL EXECUTION

Once the initialisation sequence for BOOT or USER mode has been completed, a programming kernel is resident in RAM on the H8S/2148AF, and the programming board is in transparent mode.

If the user downloads the programming kernel supplied with the EVB2148AF an ASCII protocol is setup via SCI1 of the H8S/2148AF and the Host PC. The Windows interface allows simple control of the programming kernel, or any terminal emulation program may be used to send commands. Details of the command protocol are included in the on-line help of the Windows interface.

If the user resets the EVB2148AF, or power is interrupted, the programming board is reinitialised and enters command processing (having reinitialised all control signals). Otherwise transparent operation continues until the PMODE signal is asserted.

7.7 PROGRAMMING SOFTWARE OPERATION

The Windows programming interface allows the user to simply control the programming board for operation mode transition (Normal, BOOT or USER), and then to control the programming kernel running on the H8S/2148AF.

The on-line help should be consulted for more detailed information on the operation of the interface.

The programming interface provides a project-based system where the user is able to specify the preferences for connection, and to keep track of the programming/erase cycles for each block of the FLASH memory.

Once the options have been set, the user may connect to the H8S/2148AF in the desired mode, which performs all of the operations described in Section 7.5 to enter BOOT or USER mode. The user is then able to perform Read, Erase and Program operations. For Programming the user may specify any valid S-Record file, and may specify whether verification is performed for each byte programmed.

On issuing the Disconnect command, the H8S/2148AF is reinitialised into normal execution mode.

7.8 UFPB JUMPER SETTINGS AND OPTIONS

The UFPB is provided with a number of user configurable jumpers. These allow the users to configure as required. There are two types of jumpers used, all of which are 0.1 inch pitch. The two types are as follows:

- Three-pin in-line
- Four-pin in-line

TABLE 7.3 JUMPER SETTINGS AND OPTIONS

3-Pin Jumpers

JPn	Use	Setting (1-2)	Setting (2-3)
JP4	Not Used		
JP5	MDX Control	12V/5V	12V/0V

4-Pin Jumpers

JPn	Use	Default Setting (1-3, 2-4)	Alternate Setting (1-2, 3-4)
JP1	TxD and RxD connection orientation	1:1 Connection	Crossed Connection

JPn	Use	Default Setting (1-2, 3-4)	Alternate Setting (1-3, 2-4)
JP2 & JP3	Programming Board Bypass Connection	Non-bypassed Connection	Bypassed Connction

Sections 7.9 through 7.10 describe each jumper and its alternative settings.

7.9 SERIAL PORT CONNECTIONS (JUMPERS JP1, JP2 AND JP3)

If the user wishes to use their own serial connection cable, in the case where a crossed cable is used (i.e. TxD to RxD from host to EVB and TxD to RxD from EVB to host) jumper JP1 provides the user with the capability to cross the RxD and TxD connections. See Figure 7.3 for further details.

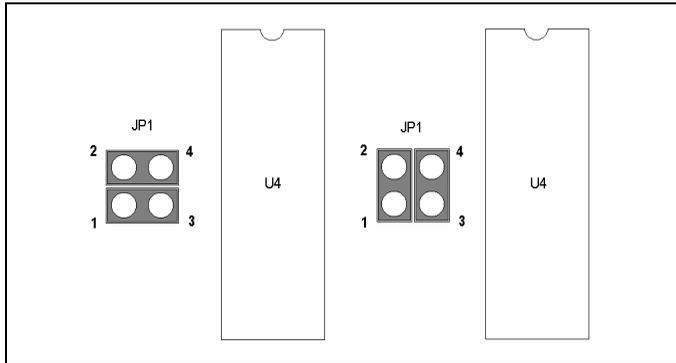


FIGURE 7.3 JP1 JUMPER SETTINGS

JP2 and JP3 allow SCI1 of the H8S/2148AF to be connected in one of three ways:

- Connected directly to the RS-232 transceiver (U4) on the UFPB (default setting)
- Connected to the SCI interface of the UFPB
- Disconnected from the UFPB

Figure 7.4 details the jumper settings for the programming board connections (if the jumpers are removed, then the SCI port is disconnected from the UFPB).

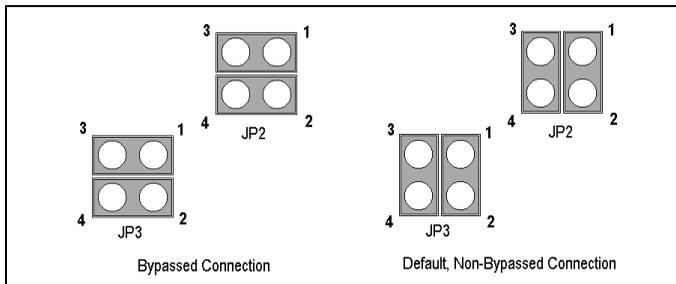


FIGURE 7.4 UFPB BYPASS CONNECTION

7.10 MDX CONTROL (JUMPER JP5)

The default option for this jumper is to have the jumper socket connected between pins 1-2. This allows the MDx signal can either be 12V and 5V depending on the mode of operation. In the alternative setting when the jumper socket is connected across pins 2-3, the MDx signal can either be 12V or 0V depending on the mode of operation