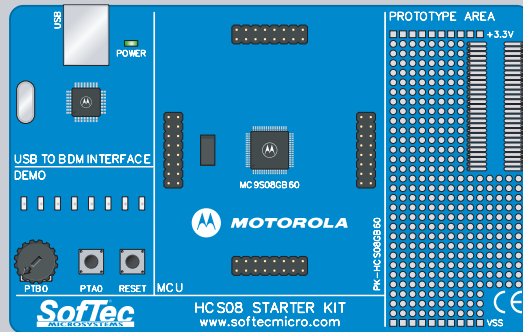# HCS08

STARTERKIT

PK-HCS08GB60
User's Manual

SofTec ®
MICROSYSTEMS

Development Tools
for the Embedded World

digital dna™

MOTOROLA
intelligence everywhere™

PK-HCS08GB60

Starter Kit for Motorola MC9S08GB60

User's Manual

SofTec ®

MICROSYSTEMS

SofTec Microsystems

E-mail (general information): info@softecmicro.com

E-mail (marketing department): marketing@softecmicro.com

E-mail (technical support): support@softecmicro.com

Web: http://www.softecmicro.com

Written by Paolo Xausa

Contents

1. Overview

1

What is the PK-HCS08GB60 Starter Kit?

The PK-HCS08GB60 Starter Kit is an entry level tool which allows you to get started with the Motorola MC9S08GB60 microcontroller.

The main features of the MC9S08GB60 microcontroller are:

- 40-MHz HCS08 CPU;
- 60 KB of FLASH memory;
- 4 KB of RAM;
- 8-channel, 10-bit Analog-to-Digital converter (ATD);
- Two serial communications interface modules (SCI);
- Serial peripheral interface module (SPI);
- Clock source options include crystal, resonator, external clock or internally generated clock with precision NVM trimming;
- Inter-integrated circuit bus module to operate up to 100 kbps (IIC);
- One 3-channel and one 5-channel 16-bit timer/pulse width modulator (TPM) modules with selectable input capture, output compare, and edge-aligned PWM capability on each channel;
- 8-pin keyboard interrupt module (KBI);
- 16 high-current pins (limited by package dissipation);
- Software selectable pull-ups on ports when used as input;
- Background debugging system;
- Input voltage range from 1.8 V to 3.6 V;

The PK-HCS08GB60 Starter Kit has been designed for the evaluation of the MC9S08GB60 microcontroller and the debugging of small user applications.

The PK-HCS08GB60 Starter Kit takes advantage of the Metrowerks CodeWarrior Integrated Development Environment (which groups an Editor, Assembler, C Compiler and Debugger)

1

and the Motorola BDM (Background Debug Module) interface, which allows the download and debug of the user application into the microcontroller's FLASH memory.

Together with CodeWarrior, PK-HCS08GB60 provides you with everything you need to write, compile, download, in-circuit emulate and debug user code. Full-speed program execution allows you to perform hardware and software testing in real time. PK-HCS08GB60 is connected to the host PC through a USB port. A prototyping area allows you to wire your own small application.

PK-HCS08GB60 offers you the following benefits:

- Real-time code execution;
- In-circuit debugging;
- In-system programming and debugging through a BDM-compatible interface;
- Demo area with push-buttons, potentiometer and user LEDs;
- Prototyping area;
- Metrowerks CodeWarrior IDE (the same user interface of all Motorola tools), with editor, assembler, C compiler and debugger.

Note: *the PK-HCS08GB60 starter kit has been designed for evaluation purposes only. Even though it has full-feature debugging options, its main limitations are:*

- *The target microcontroller is fixed (soldered to the board).*
- *The data transfer rate (PC to target and target to PC) is slow. This results in high programming times.*

For serious debugging, we suggest you to switch to the SofTec Microsystems inDART-HCS08 Series of debugging/programming tools.

1

Background Debug Module (BDM)

All MCUs in the HCS08 family contain a single-wire background debug interface which supports in-circuit programming of on-chip non-volatile memory and sophisticated non-intrusive debug capabilities. This system does not interfere with normal application resources. It does not use any user memory or locations in the memory map and does not share any on-chip peripherals. The Background Debug Module (also known as Background Debug Controller, BDC) uses a single-wire communication interface (via the BKGD line) to allow non-intrusive access to target system memory and registers.

PK-HCS08GB60 features a USB-to-BDM circuitry which allows the host PC to communicate to the microcontroller through a standard USB cable.

Note: *in this microcontroller, the BKGD line is shared with the PG0 line (same pin). The user program, therefore, must not use the PG0 line.*

PK-HCS08GB60 Board Layout

The PK-HCS08GB60 board has the following hardware features:

1. **A "USB to BDM Interface" section.** It contains the circuitry needed to electrically and logically translate BDM-like commands sent by the host PC through the USB cable to the BDM interface of the microcontroller. The PK-HCS08GB60 board is powered by the USB bus.
2. **A "Demo" section.** It features a RESET push-button, one user push-button, a potentiometer and eight user LEDs.
3. **A "MCU" section.** It contains a soldered, 64-pin MC9S08GB60 device (in QFP package) with connectors to access the I/O pins of the microcontroller for expansion prototyping. A 32.768 KHz crystal oscillator is provided.
4. **A "Prototype" section.** You can wire your own circuit here. The prototype section features both a standard, thru-hole area (for mounting traditional components) and a SMD area (for soldering SMD components in SOIC package).

1
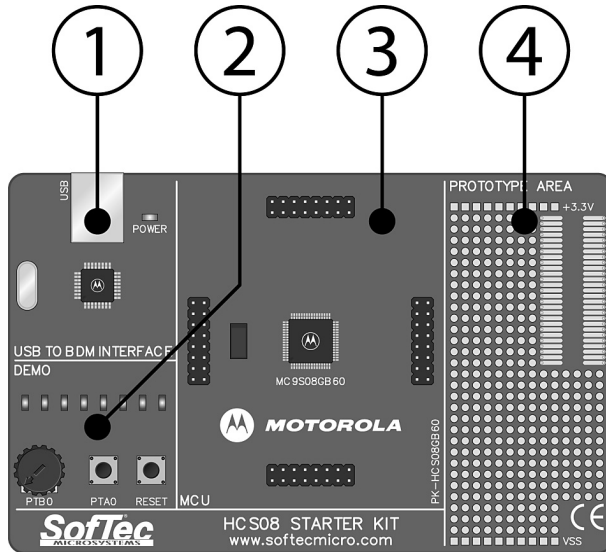
The PK-HCS08GB60 Board

CodeWarrior Integrated Development Environment

PK-HCS08GB60 comes with a free version of CodeWarrior Development Studio for HC(S)08 Microcontrollers, Special Edition.

CodeWarrior Development Studio for HC(S)08 is a powerful and easy-to-use tool suite designed to increase your software development productivity. Its Integrated Development Environment (IDE) provides unrivaled features such as Processor Expert application design

1

tool, full chip simulation, Data Visualization and project manager with templates to help you concentrate on the added value of your application.

The comprehensive, highly visual CodeWarrior Development Studio for Motorola HC(S)08 Microcontrollers enables you to build and deploy HC(S)08 systems quickly and easily. This tool suite provides the capabilities required by every engineer in the development cycle, from board bring-up to firmware development to final application development.

To use the Special Edition (4 KB code-size limited), you must have a valid license key. Without the license key the product will run in a 1 KB code-size limited demonstration mode.

To request the license key, please refer to Metrowerks website.

This documentation covers the basic setup and operation of the CodeWarrior IDE, but does not cover all of its functions. For further information, please refer to the CodeWarrior on-line help and on-line documentation provided.

Recommended Reading

This documentation describes how to use PK-HCS08GB60 together with Metrowerks CodeWarrior HC(S)08 IDE. Additional information can be found in the following documents:

- **PK-HCS08GB60 Schematic.**
- **Metrowerks' Additional Documentation**—Available from the CodeWarrior IDE.
- **Motorola HCS08 Datasheets**—Include detailed information on the devices' background debug module.

Software Upgrades

The latest version of the PK-HCS08GB60 system software is always available free of charge from our website: *http://www.softecmicro.com.* Metrowerks CodeWarrior upgrades can be found at *http://www.metrowerks.com.*

2. Getting Started

PK-HCS08GB60 Components

2

The PK-HCS08GB60 package includes the following items:

1. The PK-HCS08GB60 evaluation board;
2. A USB cable;
3. The Metrowerks CodeWarrior HC(S)08 CD-ROM;
4. The SofTec Microsystems PK-HCS08GB60 "System Software" CD-ROM;
5. A "QuickStart Tutorial" color poster;
6. This user's manual.

Host System Requirements

The PK-HCS08GB60 in-circuit debugger is controlled by an Integrated Development Environment running under Windows (CodeWarrior HC(S)08). The following hardware and software are required to run the CodeWarrior HC(S)08 user interface together with PK-HCS08GB60:

1. A 133-MHz (or higher) PC compatible system running Windows 98, Windows 2000 or Windows XP;
2. 128 MB of available system RAM plus 500 MB of available hard disk space;
3. A USB port;
4. CD-ROM drive for installation.

Installing the Software

Note: *before to connect the PK-HCS08GB60 board to the PC, it is recommended that you install all of the required software first (see below), so that the PK-HCS08GB60 USB driver will be automatically found by Windows when you connect the board.*

PK-HCS08GB60 requires that both Metrowerks CodeWarrior IDE and SofTec Microsystems PK-HCS08GB60 additional components be installed in the host PC.

Note: *Metrowerks CodeWarrior HC(S)08 IDE must be installed first. Please note that PK-HCS08GB60 only works with CodeWarrior for HC(S)08 version 3.0 or above.*

Installing Metrowerks CodeWarrior IDE

To install the CodeWarrior IDE insert the CodeWarrior CD-ROM into your computer's CD-ROM drive. A startup window will automatically appear. Follow the on-screen instructions.

Installing SofTec Microsystems Additional Components

The SofTec Microsystems additional components install all of the other required components to your hard drive. These components include:

- The PK-HCS08GB60 USB driver;
- PK-HCS08GB60 software plug-in for CodeWarrior HC(S)08;
- Examples;
- Documentation in PDF format.

To install the SofTec Microsystems additional components insert the SofTec Microsystems **"System Software"** CD-ROM into your computer's CD-ROM drive. A startup window will automatically appear. Choose **"Install Instrument Software"** from the main menu. A list of

available software will appear. Click on the **"PK-HCS08 Series Additional Components"** option. Follow the on-screen instructions.

2

Note: *if you are installing the PK-HCS08GB60 additional components on Windows 2000 or Windows XP you must have logged in as Administrator.*

Installing the Hardware

The PK-HCS08GB60 board is connected through a USB port to a host PC. Connection steps are listed below in the recommended flow order:

1. Install all the required system software as described in the previous section.
2. Insert one end of the USB cable into a free USB port.
3. Insert the other end of the USB cable into the "USB" connector on the PK-HCS08GB60 board. The green "POWER" LED on the instrument should turn on. Windows will automatically recognize the instrument and will load the appropriate USB driver.

Note: *both Windows 2000 and Windows XP may issue a warning the first time PK-HCS08GB60 is connected to the PC. This warning is related to the fact that the USB driver used by PK-HCS08GB60 is not digitally signed by Microsoft, and Windows considers it to be potentially malfunctioning or dangerous for the system. However, you can safely ignore the warning, since every kind of compatibility/security test has been carried out by SofTec Microsystems.*

Application Tutorial

This section will provide a step-by-step guide on how to launch your first PK-HCS08GB60 project and get started with the CodeWarrior HC(S)08 user interface.

The sample application reads the position of the potentiometer (connected to the microcontroller's ADC peripheral) and displays this value on the LEDs.
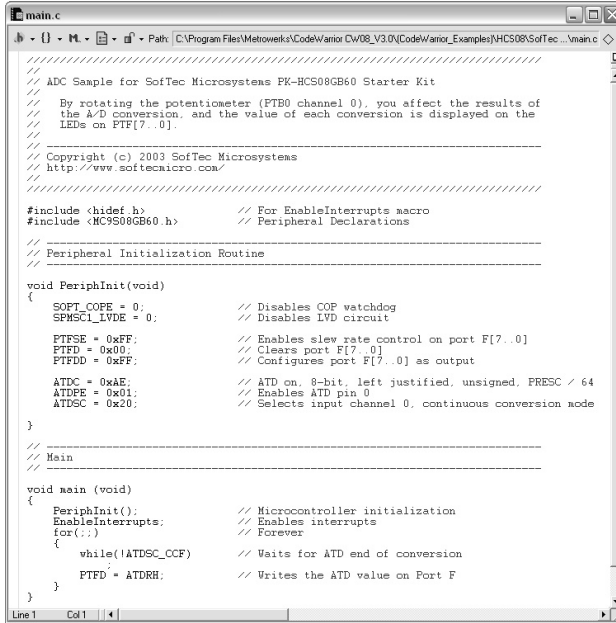
To execute the sample application, follow the next steps:

1. Ensure that the PK-HCS08GB60 board is connected to the PC (via the USB cable).
2. Start the CodeWarrior HC(S)08 IDE by selecting **Start > Programs > Metrowerks CodeWarrior > CW08 > CodeWarrior IDE**. The CodeWarrior HC(S)08 IDE will open.
3. From the main menu, choose **File > Open**. Select the **"adc.mcp"** workspace file that is located under the **"\Program Files\Metrowerks\CodeWarrior CW08\(CodeWarrior_Examples)\HCS08\SofTec Microsystems\PK-HCS08GB60\C\Adc"** directory. Click **"Open"**. The following window will appear.
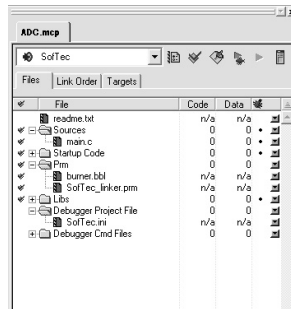
The *Project* Window

4. The C code of this example is contained in the **"main.c"** file. Double click on it to open. The following window will appear.

The Example's Source Code

5. From the main menu, choose **Project > Debug**. This will compile the source code, generate an executable file and download it to the PK-HCS08GB60 board.

6. A new debugger environment will open.

2
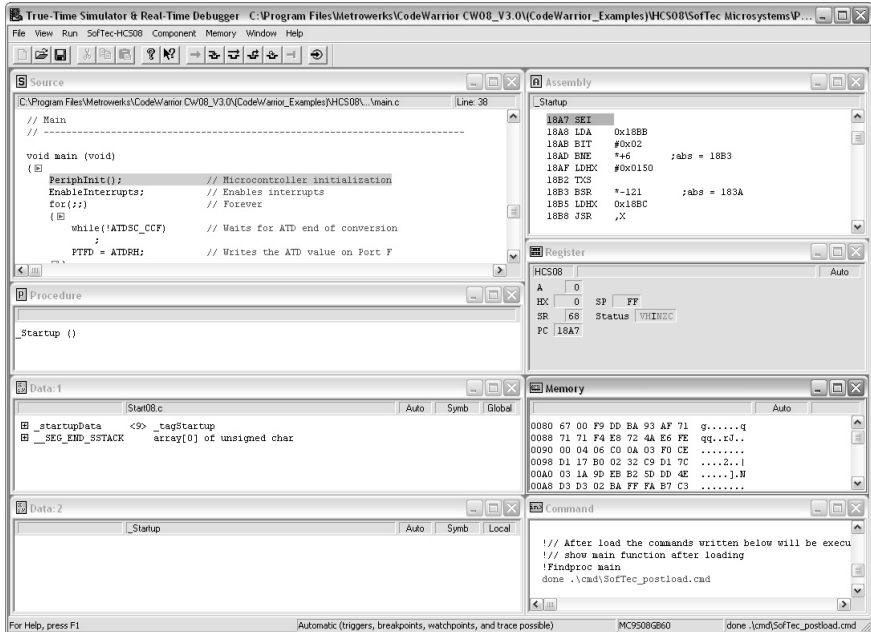
Debugging Session Started

7. From the main menu, choose **Run > Start/Continue**. The program will be executed in real-time. By rotating the potentiometer on the PK-HCS08GB60 board, you affect the results of the A/D conversion, and the value of each conversion is displayed on the LEDs.

8. From the main menu, choose **Run > Halt**. The program execution will stop. The next instruction to be executed is highlighted in the *Source* window.

9. From the main menu, choose **Run > Single Step**. The instruction highlighted in the *Source* window will be executed, and the program execution will be stopped immediately after.

10. In the *Source* window, insert a breakpoint at the **"PTFD = ATDRH;"** instruction in the **main** function. To insert the breakpoint, right-click on the **"PTFD = ATDRH;"** line and, from the pop-up menu, select **"Set Breakpoint"**.

11. Rotate the potentiometer slightly. Then, from the main menu, choose **Run > Start/Continue**. The application will restart from where it was previously stopped. The application will stop at the breakpoint location as soon as the next A/D conversion is done.

12. Issue a Single Step command (**Run > Single Step**). The new value of the A/D conversion will be displayed on the LEDs.

2

Congratulations! You have successfully completed this tutorial! You can continue to experiment with the CodeWarrior user interface and discover by yourself its potentialities. For an in-depth guide of all of the user interface features, select **Help > CodeWarrior Help** from the CodeWarrior HC(S)08 IDE's main menu.

Additional Examples

Additional examples can be found under the **"\Program Files\Metrowerks\CodeWarrior CW08\(CodeWarrior_Examples)\HCS08\SofTec Microsystems\PK-HCS08GB60"** directory.

3. Hardware Features

Introduction

PK-HCS08GB60 is an in-circuit debugger—it programs files into the MC9S08G60 microcontroller and offers debugging features like real-time code execution, stepping, and breakpoints. Its debugging features are achieved thanks to the microcontroller's integrated Background Debug Module, BDM (also referred to as Background Debug Controller, BDC).

The BDM peripheral communicates with the host PC board (via the "USB to BDM Interface" circuitry) through a dedicated, single-wire line (BKGD) of the microcontroller. The same line is also used during device programming.

Contrariwise to traditional in-circuit emulation (where the target application is executed and emulated inside the emulator), PK-HCS08GB60 uses the very same target microcontroller to carry on in-circuit execution. This means that all microcontroller's peripherals (timers, A/D converters, I/O pins, etc.) are not reconstructed or simulated by an external device, but are the very same target microcontroller's peripherals. Moreover, the PK-HCS08GB60 debugging approach ensures that the target microcontroller's electrical characteristics (pull-ups, low-voltage operations, I/O thresholds, etc.) are 100% guaranteed.

MCU Section

The "MCU" section contains the target microcontroller and the additional circuitry needed for the correct microcontroller startup. In detail:

- An MC9S08GB60 microcontroller, together with all necessary filter capacitors.
- A voltage regulator is used to shift the voltage provided by the USB bus (5 V) down to the microcontroller's operating voltage (3.3 V).
- A 32.768 KHz crystal oscillator, connected to the microcontroller's EXTAL and XTAL pins.
- A push-button in the "Demo" section is directly connected to the microcontroller's RESET pin.

3

- All of the microcontroller's pins are available on the four connectors placed around the microcontroller.

USB to BDM Interface

This section contains the circuitry needed to electrically and logically translate BDM-like commands sent by the host PC through the USB cable to the BDM interface of the microcontroller.

The USB interface is based on a Motorola MC68HC908JB16 microcontroller, which features an on-board, low-speed USB peripheral.

The USB bus provides the power supply for the board. To protect the USB bus against short circuits that may occur during experiments, the power supply circuitry features a 200 mA auto-restore fuse.

Note: even though USB specifications require low-speed devices to be used only with a captive or manufacturer-specific USB cable, we have decided to use a standard USB cable. We therefore recommend that you use the PK-HCS08GB60 board with the USB cable provided or, if you use another USB cable, ensure that the cable length does not exceed 2 meters.

Demo Section

The "Demo" section groups push-buttons, a potentiometer, and user LEDs. In detail:

- Eight user LEDs are connected to the microcontroller's Port F pins. All of the microcontroller I/O pins provide the required current to drive the low-current LEDs used by the starter kit, so no external transistors are needed.
- One user push-button, connected to the microcontroller's PTA0 pin. In order to read the status of this push-button, the microcontroller's internal pull-up must be enabled (through software) on this pin.

- One push-button connected to the microcontroller's RESET pin.
- A potentiometer, connected to the microcontroller's PTB0 pin. Please note that VREFH and VREFL (the reference voltages for the analog-to-digital converter) are connected to 3.3 V and VSS, respectively.

Prototype Area

The prototype section features both a standard, thru-hole area (for mounting traditional components) and a SMD area (for soldering SMD components in SOIC package).

3

4. Debugging Features

Creating Your Own Application

Using the Project Wizard to Create Your Application Skeleton

CodeWarrior HC(S)08 helps you get started with your own application by including a project wizard specific for HCS08-based SofTec Microsystems boards. To create a new PK-HCS08GB60 project:

1. From the main menu, select **File > New**.
2. A dialog box will appear. Select **"HC(S)08 New Project Wizard"**.
3. Follow the Project Wizard steps, making sure you select the correct microcontroller derivative you are working with and the **"SofTec Microsystems"** target connection is used.

4

Starting your first Debugging Session

The first time you enter a debugging session (by selecting **Project > Debug** from the CodeWarrior's main menu) the *MCU Configuration* dialog box will open, asking you to select the debugging hardware connected to the PC. Make sure that the hardware code is set to **"PK-HCS08GB60"**.
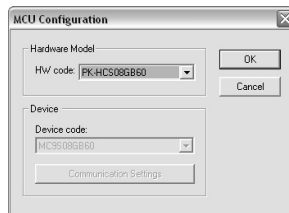
The *MCU Configuration* Dialog Box

Using Existing Projects with PK-HCS08GB60

If your project has been targeted to an emulator/simulator other than PK-HCS08GB60 and you wish to use PK-HCS08GB60 as the debugger for your project, please do the following:

1. If your project has been created with a version of CodeWarrior less than 3.0, make sure that the "target" command line option specifies the GDI target interface. To do so:

   a. Open your existing project's settings dialog box.
   b. In the **"Target Settings Panels"** section, click on the **"Build Extras"** item.
   c. If, in the debugger's argument edit box, the string "-target" appears, make sure that is set to **"-target=GDI"**.

2. CodeWarrior is interfaced to the PK-HCS08GB60 engine through a so-called "GDI interface". From the CodeWarrior HC(S)08 debugger interface, select **Component > Set Target** and choose **"HCS08"** as processor and **"GDI Target Interface"** as target interface.
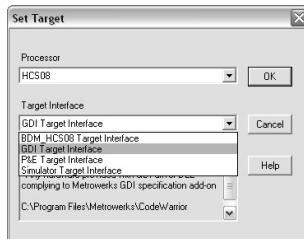
The *Set Target* Dialog Box

3. A dialog box will appear asking you to locate the GDI DLL file needed to interface with PK-HCS08GB60. Select the **SofTec_BDC08.dll** file located into the **\Program Files\Metrowerks\CodeWarrior CW08\prog** directory.
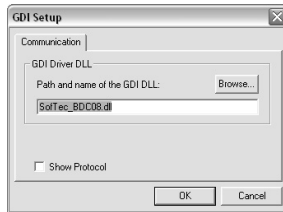
The *GDI Setup* Dialog Box

4. The *MCU Configuration* dialog box will appear allowing you to select the PK-HCS08GB60 board as the hardware debugger.
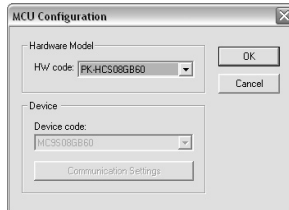
The *MCU Configuration* Dialog Box

5. On the CodeWarrior HC(S)08 debugger interface a new menu (**SofTec-HCS08**) will be created. From this menu, select **Load** and locate the object file your project is based on.

Breakpoints and Trace

CodeWarrior offers a variety of tools for analyzing the program flow: breakpoints (both simple and complex), watchpoints and a trace buffer. All these features are implemented by taking advantage of the target microcontroller's DBG peripheral.

To access breakpoints, watchpoints and trace buffer settings, open the *Trigger Module Settings* dialog box by choosing the **SofTec-HCS08 > Trigger Module Settings** menu while in debug.

When setting an instruction breakpoint, CodeWarrior does the following:

1. If available, uses the hardware address breakpoint built into the BDC module;
2. Otherwise, if possible, sets one hardware breakpoint using the DBG peripheral;
3. If no hardware breakpoints are available in the DBG peripheral (or if the trigger mode is not set to "Automatic"), a warning message is displayed, and the instruction breakpoint will not be set.

4

Note: *when setting an instruction breakpoint on a RAM location, a software breakpoint is set (the opcode present at that location is automatically replaced by the **BGND** Assembly instruction). Therefore, no hardware breakpoints are wasted.*

Note: *the Single Step command (in a C source code) and the Step Over and Step Out commands (both in a C and Assembly source code) use one hardware breakpoint.*

Notes and Tips

Entering Debug Session with CodeWarrior

When entering a debug session, the target microcontroller's FLASH memory is automatically erased, unsecured, programmed with the user application, and the trimming value is automatically calculated and programmed (to the location suggested by Motorola).

4

Note: *When programming the microcontroller with the user application (after having unsecured the device), CodeWarrior ignores (doesn't program) the security bits. As a result, when entering a debug session, the device is always unsecured, regardless of other user settings.*

Reading Peripheral Status

Care must be taken when reading some peripheral's status/data registers, since a reading operation may cause the clearing of flags. This may happen when the *Memory* window or the *Data* window is open, since these windows read microcontroller's resources during refresh operations.

Breakpoints and BGND Instruction

The BGND Assembly instruction forces the target microcontroller to enter the Active Background Debug mode, stopping program execution. CodeWarrior recognizes this event as a breakpoint and updates the contents of registers, memory, etc. Successive commands (Start/Continue, Single Step, etc.) will continue the execution of the program from the next instruction.

STOP Assembly Instruction

If the STOPE bit in the SOPT register is not set, the STOP instruction is recognized as an illegal opcode, causing the microcontroller to reset. The SOPT register can be written only once after reset.

Real-Time Memory Update

During program execution, it is possible to view/edit the contents of the *Memory* window and *Data* window in real time (edit operations are only available for RAM locations and peripheral registers). For example, it is possible to set the periodical refresh of the *Memory* window contents by choosing **Mode > Periodical** from the pop-up menu which appears by right-clicking on the *Memory* window.

FLL Usage

The host PC communicates with the microcontroller through the "USB to BDM INTERFACE" circuitry (which features an asynchronous BDM communication to the MC9S08GB60 microcontroller). The BDM communication speed depends on a clock source which, in turn, is selected by the CLKSW bit in the Status register. If the CLKSW bit is set to 1, the BDM communication clock source is the microcontroller's bus frequency; if the CLKSW bit is set to 0, the BDM communication clock source is a constant clock source (in the case of the MC9S08GB60, a 8 MHz internal clock).

PK-HCS08GB60 always sets the CLKSW bit to 0: you are therefore free to change the microcontroller's bus frequency through the FLL, since this will not affect the BDM communication.

Hardware Breakpoints and Software Breakpoints

A "hardware" breakpoint is set by taking advantage of the microcontroller's integrated debug peripheral. A hardware breakpoint doesn't waste system resources, you can set/remove them at any time (even during program execution), but the number of available hardware breakpoints is limited.

A "software" breakpoint, on the other hand, does not take advantage of the microcontroller's integrated debug peripheral.

To set a software breakpoints, there are two possibilities: you can set a software breakpoint in RAM or in FLASH. In both cases, an unlimited number of software breakpoints can be set.

To set a software breakpoint in a RAM location, just insert a breakpoint to that location: CodeWarrior will automatically replace the opcode present at that location with the **BGND** Assembly instruction.

To set a software breakpoint in a FLASH location, you have to insert the **BGND** Assembly instruction into your application's source code, recompile the code and restart a debug session. The program execution will stop as soon as the **BGND** instruction is fetched.

Advanced Debugging Features

The MC9S08GB60 microcontroller's built-in DBG module allows you to set "complex" hardware breakpoints and to take advantage of a trace buffer.

While in debug, under the **SofTec-HCS08** menu you will find the **"Trigger Module Settings…"** command, that opens a dedicated dialog box which allows you to handle all of the parameters of the microcontroller's debug peripheral.

4

DataBlaze Programming Utility

A full-featured programming utility (DataBlaze) is also provided with the PK-HCS08GB60 Starter Kit. To start the DataBlaze utility select **Start > Programs > SofTec Microsystems > PK-HCS08 Series > DataBlaze Programmer**.

DataBlaze offers the following advanced features:

- Code memory editing;
- Blank check/erase/program/verify/read operations;
- Project handling;
- One-button, multiple-operations programming ("Auto" feature);
- Automatic trimming calibration;
- Serial numbering.

Note: *due to the evaluation purposes of the PK-HCS08GB60 starter kit (and therefore to the slow data transfer rate from the PC to the target and vice versa), the DataBlaze programming utility takes a long time to write to/read from the whole microcontroller memory.*

For serious debugging/programming, we suggest you to switch to the SofTec Microsystems inDART-HCS08 Series of debugging/programming tools.

4
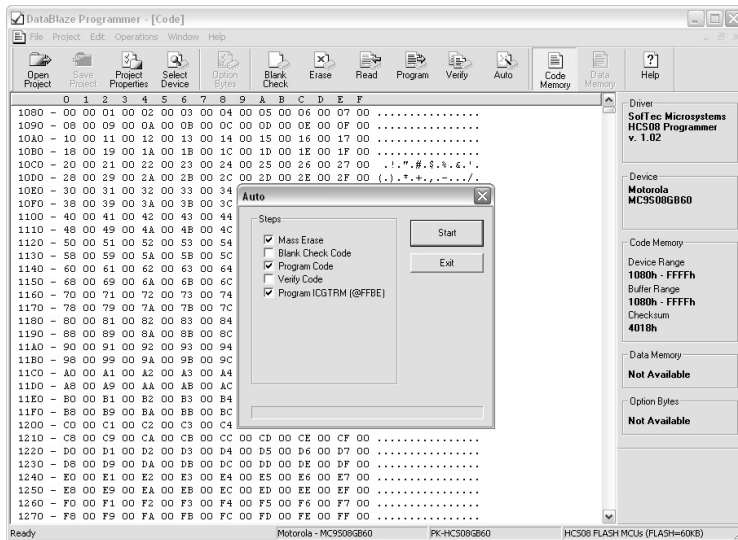
The DataBlaze User Interface

DataBlaze Notes

- The "Mass Erase" operation always blanks the device (even if the device is protected or secured) and "unsecures" the device (the NVOPT FLASH register is programmed with 0xFE).
- The "Blank Check" operation doesn't blank check the NVOPT FLASH register.
- The "Program" operation automatically verifies the programmed data, by reading back the programmed data and checking it against the buffer sent to the target device. The "Verify" operation is much more secure (but slower), since it reads back the programmed data and checks it against the data buffer present in the host PC.
- In case of verifying error, please verify the value programmed to the FLASH Options/Security Byte location. The bit 0 of this byte is always programmed to 0, so any attempt to program it to 1 will cause a verifying error.
- The "Read", "Program" and "Verify" operations are performed (when possible) by setting the target microcontroller's FLL peripheral so that the maximum BDM communication speed is achieved.
- The FLASH location suggested by Motorola where to store the trimming value is reserved by DataBlaze. DataBlaze automatically calculates the trimming value and writes it to that location.

4

5. Troubleshooting

Common Problems and Solutions

This section reports some common problems that may arise during general use.

Communication Can't Be Established with PK-HCS08GB60

1. Make sure the PK-HCS08GB60 starter kit is connected to the PC and powered on. PK-HCS08GB60 is powered by the USB connection.
2. If you connected the PK-HCS08GB60 board to the PC before installing the CodeWarrior user interface and the SofTec Microsystems Additional Components, the PK-HCS08GB60 USB driver may not have been correctly installed on your system. Unplugging and replugging the USB cable is of no use, since Windows has marked the device as "disabled". As a consequence, CodeWarrior cannot communicate with the PK-HCS08GB60 board.
To restore the USB driver (provided both CodeWarrior and SofTec Microsystems Additional components have been installed), perform the following steps under Windows XP:

   - Plug the PK-HCS08GB60 board to the PC.
   - Open the Control Panel (**Start > Settings > Control Panel**).
   - Open the **"System"** options.
   - Select the **"Hardware"** tab.
   - Click the **"Device Manager"** button.
   - The **"PK-HCS08 Series Starter Kit"** device will be shown with an exclamation mark next to it. Double click on this device.
   - In the **"General"** tab, click the **"Reinstall Driver"** button. Follow the on-screen instructions.

5

3. Make sure you are working with the correct hardware model. To view/change the hardware model in use, choose **SofTec-HCS08 > MCU Configuration** from the CodeWarrior HC(S)08 debugger's main menu.

4. If the **SofTec-HCS08** menu is not present in the CodeWarrior HC(S)08 debugger's main menu, this is because the target has not been recognized by CodeWarrior ("No link to Target" appears in the status bar). In this case, from the **GDI** menu, choose **MCU Configuration** and verify that the hardware code is set correctly.

Stepping Execution is Slow

When the *Memory* window is open, step commands may execute slower, since the *Memory* window contents need to be refreshed after every step.

STOP Assembly Instruction Causes a Microcontroller Reset

If the STOPE bit in the SOPT register is not set, the STOP instruction is recognized as an illegal opcode, causing the microcontroller to reset. The SOPT register can be written only once after reset.

Communication Lost During Debugging

This problem may occur because:

1. The BKGDPE bit in the SOPT (System Option Register) register has been set to 0, configuring the BKGD pin as a generic I/O pin. The BKGD pin must be reserved for the BDM communication (BKGDPE = 1).

2. A microcontroller reset has occurred. Among other things, this may be caused by the COP peripheral. After reset, the COP is enabled, so your program must either disable it or reset its timer.

Getting Technical Support

Technical assistance is provided free to all customers. For technical assistance, documentation and information about products and services, please refer to your local SofTec Microsystems partner.

SofTec Microsystems offers its customers a free technical support service at *support@softecmicro.com*. Before getting in contact with us, we advise you to check that you are working with the latest version of the PK-HCS08GB60 system software (upgrades are available free of charge at *http://www.softecmicro.com*). Additional resources can be found on our HCS08 online discussion forum.

5

Appendix A. Electrical and Physical Specifications

Operating Voltage	4.75 to 5.0 V DC (provided by the USB connection)
Power Consumption	200 mA (max)
Dimensions	137 x 86 x 15 mm
Weight	55 g
Operating Temperature	0 °C to 50 °C
Storage Temperature	-20 °C to 70 °C
Humidity	90% (without condensation)

Electrical and Physical Specifications

A