

inDART[®] - HC08
In-Circuit
Debugger/Programmer
for Freescale
HC08 Family FLASH
Devices

User's Manual
Rev. 2.0



Copyright © 2006 SofTec Microsystems®

DC01027

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>

Important

SofTec Microsystems reserves the right to make improvements to the inDART® Series In-Circuit Debuggers, their documentation and software routines, without notice. Information in this manual is intended to be accurate and reliable. However, SofTec Microsystems assumes no responsibility for its use; nor for any infringements of rights of third parties which may result from its use.

SOFTEC MICROSYSTEMS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

Trademarks

SofTec Microsystems is a registered trademark of SofTec Microsystems, Spa.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation.

PC is a registered trademark of International Business Machines Corporation.

Other products and company names listed are trademarks or trade names of their respective companies.

Written by Paolo Xausa

Contents

1. Overview	5
What is inDART-HC08?	5
What is In System Programming (ISP)?	6
CodeWarrior Integrated Development Environment	7
SofTec Microsystems' DataBlaze Programming Utility	8
Demo Boards	8
Recommended Reading	8
Software Upgrades	9
2. Getting Started	11
inDART-HC08 Components	11
Host System Requirements	12
Installing the Software	12
<i>Installing the CodeWarrior IDE</i>	12
<i>Installing SofTec Microsystems' Additional Components</i>	12
Installing the Hardware	13
Application Tutorial	15
3. inDART-HC08 Operations	21
inDART-HC08 Working Principles	21
Limitations	21
Configuring a Debugging Session	22
<i>Creating Your Own Application</i>	22
<i>MCU Configuration</i>	24
<i>MCU Configuration > Communication Settings</i>	24
<i>MCU Configuration > Communication Settings (Communication)</i>	25
<i>MCU Configuration > Communication Settings (MON08)</i>	26
<i>MCU Configuration > Communication Settings (Power Supply)</i>	28
<i>Using Existing Projects with inDART-HC08</i>	28
Notes and Tips	29
<i>Stop Command Handling</i>	29
<i>Breakpoints and Swi Instruction</i>	30
<i>Reading Peripheral Status</i>	30
<i>Interrupt Execution during Steps</i>	30
<i>Peripheral Status during Steps</i>	30

<i>Internal RC Oscillator Calibration</i>	30
DataBlaze Programming Utility	30
SofTec Microsystems' inSIDE HC08 Debugger	31
4. Troubleshooting	33
Common Problems and Solutions	33
<i>Communication Can't Be Established with inDART-HC08</i>	33
<i>A Communication Error Is Returned on a Program Execution Command (Start/Continue, Single Step, etc.)</i>	34
<i>Peripheral Speed is Low</i>	34
<i>Stepping Execution is Slow</i>	34
Diagnostic Test	34
Getting Technical Support	35
Appendix A. Electrical and Physical Specifications	37
Appendix B. MON08 Connector	39
In-Circuit Debugging	39
Standalone Target Operation	39

1. Overview

What is inDART-HC08?

inDART-HC08 is a powerful entry-level tool for Freescale HC08-based systems. inDART-HC08 takes advantage of the CodeWarrior HC08 Integrated Development Environment and the ISP (In-System Programming) feature to program the FLASH memory of the HC08 family of microcontrollers. Together with CodeWarrior HC08, inDART-HC08 provides you with everything you need to download (program), in-circuit emulate and debug user code. Full speed program execution allows you to perform hardware and software testing in real time. inDART-HC08 is connected to the host PC through a USB port, while the 16-pin connector of the product fits into the target's standard MON08 connector. On Design Kit packages, a full-featured experiment board for a specific HC08 microcontroller is also included.

inDART-HC08 offers you the following benefits:

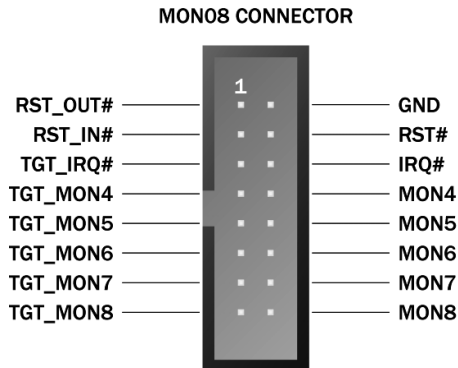
- Real-time code execution without probes—works with all packages (MON08-compatible ISP connector);
- In-circuit debugging;
- Built-in FLASH programmer (DataBlaze programming utility);
- 2.2 to 5.0 V devices supported;
- Standard chip used—no bondouts, 100% electrical characteristics guaranteed;
- Working frequency up to the microcontroller's maximum;
- Jumperless hardware mode setting;
- Automatic V_{TST} voltage generation;
- Automatic target baud rate detection;
- Hardware self diagnostic test;
- USB connection to the PC;
- CodeWarrior IDE (the same user interface of all Freescale tools), with editor, assembler, C compiler and debugger.

1

What is In System Programming (ISP)?

The ISP feature allows you to update the content of FLASH program memory when the chip is already plugged on the application board. The ISP feature can be implemented with a minimum number of added components and board area impact.

inDART-HC08 uses a standard, 16-pin ISP connector (MON08-compatible) to program the target device. You must therefore provide such connector (see the diagram below) on your target board. The MON08 connector is the same connector used for debugging tools such as Freescale ICS08 boards. Throughout this documentation, MON08 connector is used as a synonym of ISP connector.



Pin Name	Pin #	Pin #	Pin Name
RST_OUT#	1	2	GND
RST_IN#	3	4	RST#
TGT_IRQ#	5	6	IRQ#
TGT_MON4	7	8	MON4
TGT_MON5	9	10	MON5
TGT_MON6	11	12	MON6
TGT_MON7	13	14	MON7
TGT_MON8	15	16	MON8

MON08 Signals

- **RST_OUT#:** Reset signal to target system: GND or open drain output reflecting the state of the MCU RST# and RST_IN# signals.
- **GND:** System ground.
- **RST_IN#:** Reset signal from target system: GND to V_{DD} input to control the state of the MCU RST# and RST_OUT# signals.
- **RST#:** MCU reset; held at V_{PP} (or V_{DD} , depending on the target microcontroller) out of reset. No other target-system logic should be tied to this signal.
- **TGT_IRQ#:** Interrupt signal from target system: GND to V_{DD} input to control the state of the MCU IRQ# signal.
- **IRQ#:** MCU interrupt; held at V_{PP} when the TGT_IRQ# signal is not asserted.
- **TGT_MONX:** I/O pins connected to target application.
- **MONX:** I/O pins connected to target microcontroller.

Each of the MON08 connector' lines must be connected to the appropriate pins of the specific target microcontroller used. The CodeWarrior HC08 IDE, once a target microcontroller has been selected, automatically shows you how to connect that specific device to the MON08 connector.

CodeWarrior Integrated Development Environment

inDART-HC08 comes with a free version of CodeWarrior Development Studio for HC08 Microcontrollers, Special Edition.

The CodeWarrior Development Studio for Freescale HC08 Microcontrollers enables you to build and deploy HC08 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. With a common, project-based, development environment reuse becomes a natural by-product as each team builds on the work already completed by the previous team.

CodeWarrior Development Studio for HC08 Microcontrollers, Special Edition, includes the CodeWarrior integrated development environment (IDE); 16 KB code-size limited C compiler and C source-level debugger; macro assembler and Assembly-level debugger and FLASH programming support. The Special Edition allows you to evaluate CodeWarrior Development Studio for HC08 Microcontrollers at no cost.

To use the Special Edition, 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 log on Metrowerks' website at <http://www.metrowerks.com/MW/Secure/Eval/HC08/default.htm> (the installer already has installed the shortcut "Get HC08 Special Edition License" which points to the URL above).

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

SofTec Microsystems' DataBlaze Programming Utility

The SofTec Microsystems "**System Software**" CD-ROM contains the DataBlaze programming utility targeted to the inDART-HC08 board.

DataBlaze is a full-featured programming utility which offers a complete set of programming features like memory editing, blank check/erase/verify operations and read/write operations.

Demo Boards

On Design Kit packages, a full-featured, microcontroller-specific experiment board is also included. The demo board can be used for evaluation/experiments in the absence of a target application board. All demo boards feature a ZIF socket for easy microcontroller replacement—demo boards, used in conjunction with inDART-HC08, can be used as programmers.

Recommended Reading

This documentation describes how to use inDART-HC08 together with CodeWarrior HC08 IDE and DataBlaze programming environment. Additional information can be found in the following document:

- **Evaluation Board-Specific Information**—If you bought a Design Kit, additional documentation explains evaluation board's details. This

documentation is available under **Start > Program Files > SofTec Microsystems > User's Manuals**.

- **CodeWarrior Additional Documentation**— Available from the CodeWarrior IDE.
- **SofTec Microsystems' inSIDE HC08 Debugger**—User's Manual of the SofTec Microsystems' alternative debugger provided for free together with inDART-HC08. This documentation is available under **Start > Program Files > SofTec Microsystems > User's Manuals**.
- **Freescale HC08 Datasheets**—Include detailed information on the devices' monitor entry configurations.
- **CPU08 Central Processor Unit Reference Manual**—Programming reference containing the description of the full HC08 instruction set.

Software Upgrades

The latest version of the inDART-HC08 system software is always available for free on our website: <http://www.softecmicro.com>. CodeWarrior upgrades can be found at <http://www.metrowerks.com>.

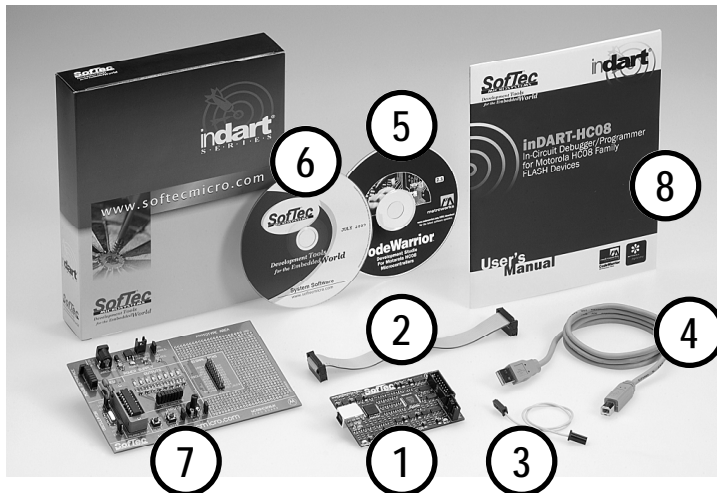
2. Getting Started

2

inDART-HC08 Components

The inDART-HC08 package includes the following items:

1. The inDART-HC08 in-circuit debugger/programmer unit;
2. A 20-cm, 16-conductor ISP cable;
3. A 30-cm, 1-conductor “VDD OUT” cable;
4. An USB cable;
5. The CodeWarrior HC08 CD-ROM;
6. The SofTec Microsystems’ inDART-HC08 “System Software” CD-ROM;
7. A full-featured, microcontroller-specific experiment board (on Design Kit packages only).



inDART-HC08 Components

2

Host System Requirements

The inDART-HC08 in-circuit debugger is controlled by an Integrated Development Environment running under Windows (CodeWarrior HC08). The following hardware and software is required to run the CodeWarrior HC08 user interface:

1. A 133-MHz (or higher) PC compatible system running Windows 98, Windows Me, Windows 2000 or Windows XP;
2. 128 MB of available system RAM plus 600 MB of available hard disk space;
3. An USB port;
4. CD-ROM drive for installation.

Installing the Software

inDART-HC08 requires that both the CodeWarrior IDE and SofTec Microsystems' inDART-HC08 additional components are installed in the host PC.

Note: *the CodeWarrior HC08 IDE must be installed first.*

Installing the 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 inDART-HC08 USB driver;
- inDART-HC08 software plug-in for CodeWarrior HC08 (complete of examples, stationeries, etc.);

- SofTec Microsystems' DataBlaze programming utility;
- SofTec Microsystems' inSIDE HC08 debugger (a tiny alternative to CodeWarrior HC08);
- 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 "**inDART-HC08 Additional Components**" option. Follow the on-screen instructions.

Note: *if you are installing the inDART-HC08 additional components on Windows 2000 or Windows XP you must have logged in as Administrator.*

Installing the Hardware

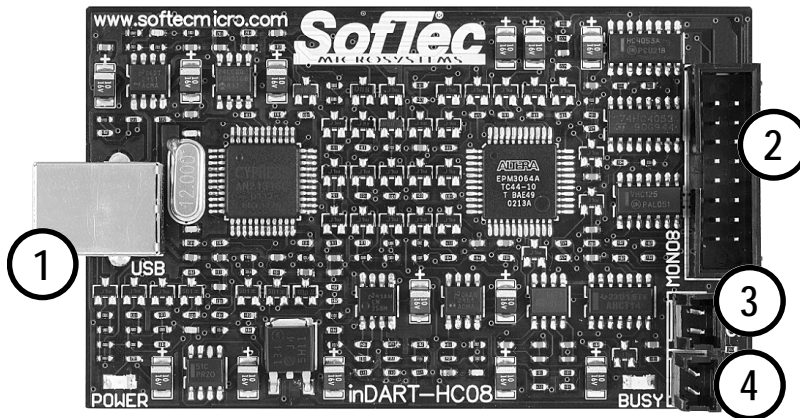
The inDART-HC08 in-circuit debugger is connected through an USB port to a host PC. Connection steps are listed below in the recommended flow order:

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 inDART-HC08 board. The green "POWER" LED on the instrument should turn on. Windows will automatically recognize the instrument and will load the appropriate USB driver.
4. Insert one end of the ISP cable into the "MON08" connector on the inDART-HC08 board.
5. Insert the other end of the ISP cable into the MON08 connector of the demo board or target application.
6. Power the demo board or target application. When using a SofTec Microsystems' demo board, there are two power options:
 - a. **Manual Power Control:** power is supplied through the "UNREG. VDD" connector (9-12 V DC, 200 mA wall plug-in power

supply with a 2.1 mm pin and sleeve plug with positive in the center and sleeve as ground). The “VDD SOURCE” jumper must select the “UNREG.” option. This power option requires you to manually turn power supply on/off during debugging/programming.

- b. **Automatic Power Control:** power is supplied through the “REG. VDD” connector. The “VDD SOURCE” jumper must select the “REG.” option. This power option allows the inDART-HC08 board to automatically handle the demo board’s V_{DD} line for debugging/programming. The “VDD OUT” cable must be inserted into the “VDD OUT” connector on the inDART-HC08 board and into the “REG. VDD” connector on the demo board.

Please refer to the following figure to get an overview of the inDART-HC08 unit.



1. USB Connector
2. MON08 Connector
3. “VDD OUT” Connector
4. “OSC OUT” Connector

The inDART-HC08 Unit

Note: *both Windows 2000 and Windows XP may issue a warning the first time inDART-HC08 is connected to the PC. This warning is related to the fact that the USB driver used by inDART-HC08 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. Additionally, under Windows XP, the “Found New Hardware Wizard” procedure may occur twice.*

Application Tutorial

This section will provide a step-by-step guide on how to launch your first inDART-HC08 project and get started with the CodeWarrior HC08 user interface.

Note: *the example provided assumes that inDART-HC08 is used with the IDB-HC08JL demo board. If you have a different demo board, some procedures may differ slightly. This tutorial is based on an Assembly example. Additional examples in C are also provided. For evaluation board-specific features, please refer to the evaluation boards' PDF manuals installed by SofTec Microsystems' additional components.*

The sample application configures the A/D peripheral to convert on the A/D channel connected to the potentiometer and displays the results on the LEDs.

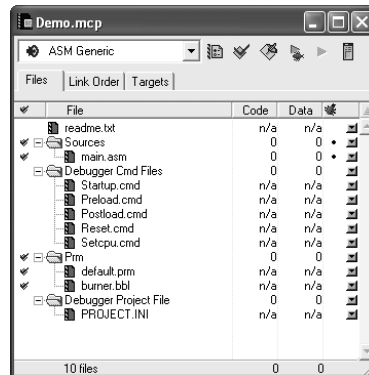
1. Ensure that the inDART-HC08 board is connected to the PC (via the USB cable) and that the demo board is powered on via the “REG. VDD” connector (the “VDD OUT” cable must be connected and the “VDD SOURCE” jumper must select the “REG.” option).
2. Ensure that the inDART-HC08 board is connected to demo board via the MON08 connector. In order to work with inDART-HC08, the jumpers in the evaluation board's MON08 connector must be removed, and the MON08 cable of inDART-HC08 must be connected to the evaluation

2. Getting Started

2

board's MON08 connector (J1), taking care of the proper polarity. The evaluation board's J2 connector is not used in this mode.

3. Make sure that the provided microcontroller sample is correctly inserted into its evaluation board socket.
4. Make sure that the evaluation board's oscillator selection jumpers select the on-board oscillator.
5. Make sure that all of the evaluation board's "LED ENABLE" jumpers and the "POTENTIOMETER ENABLE" jumper are inserted.
6. Start the CodeWarrior HC08 IDE by selecting **Start > Programs > Metrowerks > CodeWarrior CW08 > CodeWarrior IDE**. The CodeWarrior HC08 IDE will open.
7. From the main menu, choose **File > Open**. Select the "**Demo.mcp**" workspace file that is located under the "**\Program Files\Metrowerks\CodeWarrior\CodeWarrior_Examples\HC08\SofTec Microsystems\inDART-HC08\J13\Asm**" directory. Click "**Open**". The following window will appear.



The *Project Window*

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


```

main.asm
Path: C:\Program Files\Metrowerks\CodeWarrior Cw08_V2.1\CodeWarrior_Examples\...\main.asm
;*****
;      MAIN ASM
;*****
; Copyright (c) 2002 SofTec Microsystems
; http://www.softemicro.com/
;*****
;
; ADC Sample for SofTec Microsystems' IDB-HC08JL Demo Board
;
; This program configures the A/D peripheral to convert on A/D channel 3
; (PTB3, connected to the potentiometer) and displays the results on the LEDs
; on PTD. Make sure that all of the "LED ENABLE" jumpers and the
; "POTENTIOMETER ENABLE" jumpers are inserted.
;*****
;*****
; XDEF Entry, irq_isr, main
;*****
; Peripheral registers definition
;*****
PTD    EQU $0003    ; Port D data register
DDRD   EQU $0007    ; Port D data direction register
CONFIG1 EQU $001F    ; Configuration register 1
ADSCR  EQU $003C    ; ADC status and control register
ADR    EQU $003D    ; ADC data register
;*****
DEFAULT_ROM SECTION
;*****
; Peripheral Initialization
;*****
init:
    bset 0, CONFIG1    ; Disables COP
    mov  $FF, DDRD     ; Configures port D as output
    mov  $23, ADSCR    ; Enables ADC channel 3
    rts
;*****
Line 55 Col 10

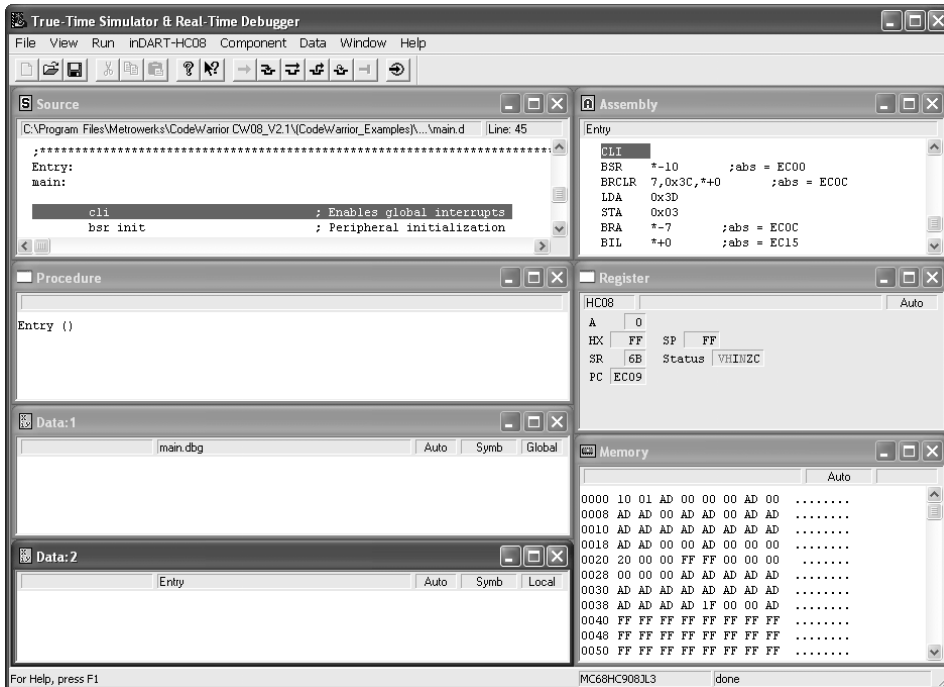
```

The Example's Source Code

9. From the main menu, choose **Project > Debug**. This will generate an executable file and will download it to the inDART-HC08 board.
10. A new debugger environment will open. In the *Source* window, the first line to be executed is highlighted.

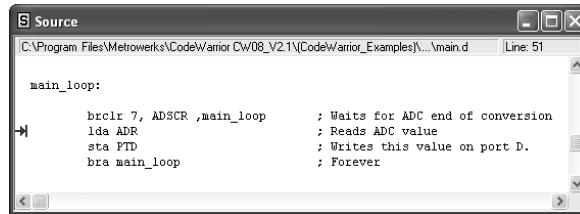
2. Getting Started

2



Debugging Session Started

11. From the main menu, choose **Run > Start/Continue**. The program will be executed in real-time. Please note that the “BUSY” LED on the inDART-HC08 unit turns on. By rotating the potentiometer on the demo board, you affect the results of the A/D conversion, and the binary value of each conversion is displayed on the LEDs.
12. From the main menu, choose **Run > Halt**. The program execution will stop. The next instruction to be executed is highlighted in the *Source* window.
13. 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.
14. In the *Source* window, insert a breakpoint at the “**lda ADR**” instruction in the **main_loop** subroutine. To insert the breakpoint, right-click on the “**lda ADR**” line and, from the pop-up menu, select “**Set Breakpoint**”.



```
Source
[C:\Program Files\Metrowerks\CodeWarrior\CW08_V2.1\CodeWarrior_Examples\...\main.d] Line: 51

main_loop:
    brclr 7, ADSCR ,main_loop    ; Waits for ADC end of conversion
    lda ADR                      ; Reads ADC value
    sta PTD                      ; Writes this value on port D.
    bra main_loop                ; Forever
```

Breakpoint Set

15. 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.
16. From the main menu, choose **Run > Single Step**. The “**lda ADR**” instruction will be executed. In the *Registers* window, you can see how that instruction affected the value contained on the Accumulator. This value is the result of the A/D conversion.
17. Issue another Single Step command (**Run > Single Step**). The Accumulator value will be displayed on the LEDs.

Congratulations! You have successfully completed this tutorial! You can continue to experiment with the CodeWarrior user interface and discover by your own its potentialities. For an in-depth guide of all of the user interface features, select **Help > CodeWarrior Help** or **Help > Online Manuals** from the CodeWarrior HC08 IDE's main menu.

3. inDART-HC08 Operations

inDART-HC08 Working Principles

inDART-HC08 is an in-circuit debugger as well as a programming tool. It programs files into the HC08 microcontrollers and offers debugging features like real-time code execution, stepping, and breakpoint. Its debugging features are achieved thanks to the microcontroller's integrated monitor firmware and Break Module.

The integrated monitor communicates with the host PC through a bidirectional, command-based protocol via some dedicated lines of the microcontroller (which are therefore reserved during debugging sessions). The same lines are also used during device programming.

Contrariwise to traditional in-circuit emulation (where the target application is executed and emulated inside the emulator), inDART-HC08 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 inDART-HC08 debugging approach ensures that the target microcontroller's electrical characteristics (pull-ups, low-voltage operations, I/O thresholds, etc.) are 100% guaranteed. The trade-off, however, is that the target microcontroller must be properly configured and ready to execute target applications.

Limitations

Since inDART-HC08 is based on the in-chip debugging features of the HC08 family of microcontrollers, some hardware and software limitations apply. The main ones are listed below; for the complete list of limitations please refer to the microcontroller's data sheet.

- The pin dedicated to the host communication is reserved—in particular, the corresponding bit in the Data Direction Register must not be changed (must be left to input);

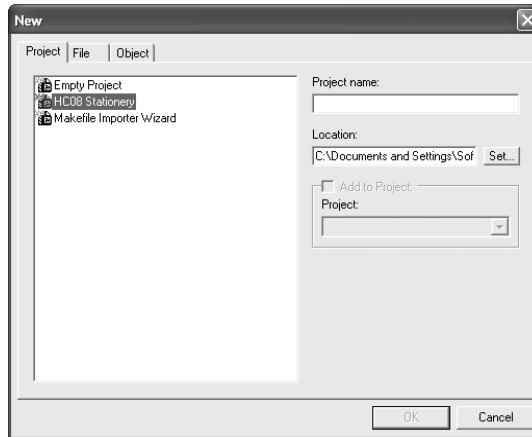
- The Break Module peripheral is reserved, and only one hardware breakpoint is available—however, you can insert a “swi” instruction into your code to generate a software breakpoint;
- The “swi” instruction is reserved and can be only used to generate a software breakpoint;
- Step commands which involve the execution of two or more Assembly instructions waste one hardware breakpoint—therefore, if one breakpoint was already set by the user, the step command cannot be executed;
- The “Halt” debugging command (in the CodeWarrior HC08 user interface) will not work unless the IRQ interrupt is properly handled;
- 13 bytes of stack are wasted by the on-chip monitor—therefore the addresses from SP-13 to SP are reserved;
- Registers which affect the target microcontroller’s clock speed must not be changed during debugging sessions.

Configuring a Debugging Session

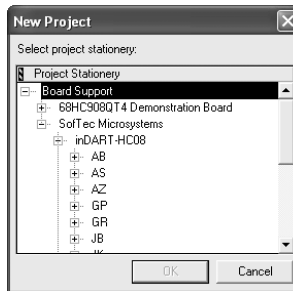
Creating Your Own Application

CodeWarrior HC08 helps you getting started with your own application by including a project stationery specific for the inDART-HC08 board. To create a new inDART-HC08 project:

1. From the main menu, select **File > New**. The following dialog box will appear:

The *New Project* Dialog Box

2. Select “**HC08 Stationery**” as project type and give the project a name. Click “**OK**”. The following dialog box will appear:



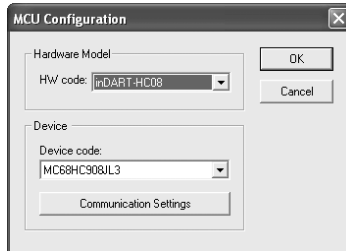
Project Stationery Selection

3. Two inDART-HC08 emulator’s project stationeries (for each HC08 microcontroller derivate) are available: one for the Assembly language, one for the C language. They are located under **Board Support > Softec Microsystems > inDART-HC08**.
4. After selecting the appropriate stationery, an empty project will be available for you to work with.

MCU Configuration

The first time you start the debugging of your application (with the command **Project > Debug**), the *MCU Configuration* dialog box will appear, allowing you to select the hardware model and the target microcontroller you are working with.

3

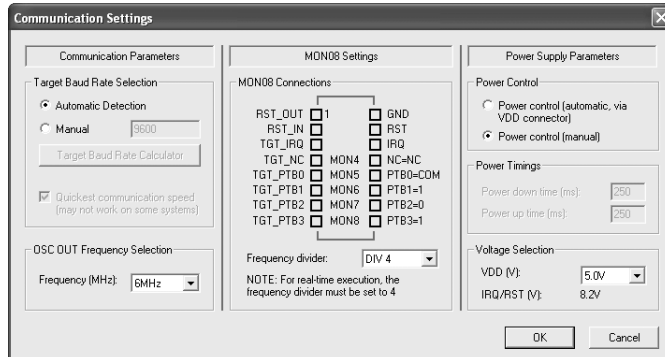


The *MCU Configuration* Dialog Box

First, ensure that the **“Hardware Model”** parameter is set to **“inDART-HC08”**. Then, set the **“Device Code”** parameter to the specific target microcontroller you are working with.

MCU Configuration > Communication Settings

The **“Communication Settings”** button in the *MCU Configuration* dialog box allows you to fine-tune critical parameters needed for proper operation with the chosen target microcontroller.



The *Communication Settings* Dialog Box

The dialog box is divided into three sections: “Communication Parameters”, “MON08 Settings” and “Power Supply Parameters”. All of the parameters must be carefully set, otherwise unsuccessful operations will result.

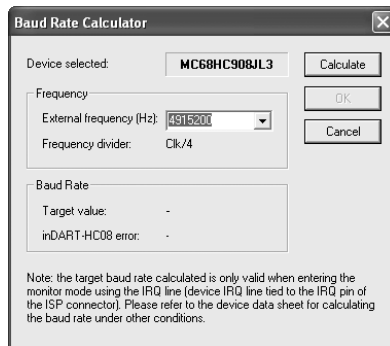
MCU Configuration > Communication Settings (Communication)

Communication parameters include the baud rate to be used to communicate with the target board. inDART-HC08 has the capability of automatically determine the correct target baud rate in the most common situations; the “Automatic Detection” is the default option. However, if for some reasons the automatic detection doesn’t work, you still have the possibility to select the baud rate manually.

3

Note: *when using the manual baud rate selection, keep in mind that the correct target baud rate value to use critically depends on the target hardware; different target configurations require different baud rate values. It is highly recommended that you consult the data sheet of the target microcontroller (the section about entering the monitor mode) in order to calculate the correct baud rate to set. An improper target baud rate value may result in programming errors.*

As an help, a baud rate calculator is provided. Just click on the “**Target Baud Rate Calculator**” button and enter the microcontroller external clock frequency and the appropriate baud rate will be calculated for you. Please note, however, that this method only works if the IRQ line of the microcontroller is tied to the IRQ line of the MON08 connector. In all other cases, again, you must consult the data sheet of the target microcontroller.



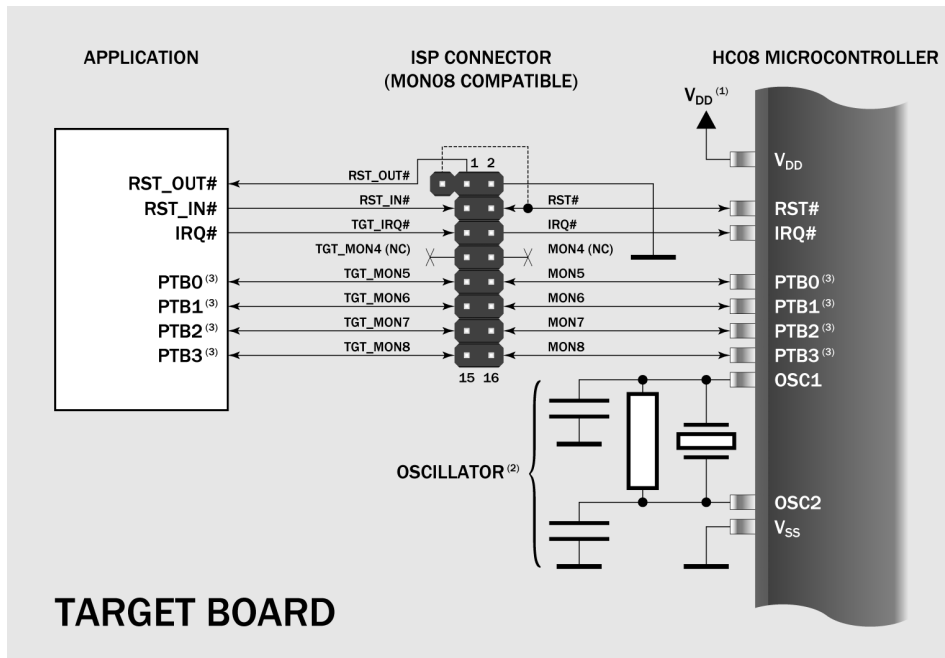
The *Baud Rate Calculator* Dialog Box

On some devices it is possible to enable a fast data transfer rate. It is important to understand that this communication mode does not affect the baud rate value used to enter the monitor mode (specified above), but the speed at which programming data is transferred from the PC to the target system **after** entering the monitor mode. To enable the fast data transfer mode, check the **“Quickest Communication Speed”** check box. Under some extreme circumstances, however, this communication mode may not work. Uncheck the option if necessary.

The **“OSC OUT Frequency Selection”** parameter allows you to program the frequency available at the “OSC OUT” connector in the inDART-HC08 board. The OSC OUT signal can be used as an auxiliary clock source, and is particularly useful when the target microcontroller requires an external clock which is not otherwise available on the target board.

MCU Configuration > Communication Settings (MON08)

This section shows you what lines of the target microcontroller must be tied to the MON4-MON8 lines of the MON08 connector. Depending on the microcontroller, either the MON4 or MON5 line is dedicated to the single-wire communication (which is reserved during debugging). The other MON lines must be driven at specific values at reset (and are available for the target application afterwards). inDART-HC08 automatically drives all of the required lines upon reset, and release them soon afterwards (exception made for the single-wire communication line which is always reserved). Please note that each microcontroller implements the MON08 interface on different ports and pins. The following diagram illustrates a typical MON08 connection.



Typical MON08 Connections for a MC68HC908JL3 Target Board

Note⁽¹⁾: the V_{DD} line needs to be driven correctly at reset. inDART-HC08 can automatically handle this line if the “VDD OUT” cable is used

Note⁽²⁾: the inDART-HC08’s OSC OUT signal can be used instead as an auxiliary clock source, and is particularly useful when the target microcontroller requires an external clock which is not otherwise available on the target board.

Note⁽³⁾: each microcontroller implements the MON08 interface on different ports and pins. MON4-MON8 lines must be therefore tied to the appropriate pins of the specific target microcontroller.

The “**Frequency divider**” parameter should always be set to 4 in order to guarantee that, in monitor mode, all of the target microcontroller’s peripherals

run at the same speed they do in user mode. If, however, you need to speed up program execution, you can set this parameter to 2, therefore doubling the microcontroller's speed in monitor mode. Note, however, that not all peripherals will work at this doubled speed.

MCU Configuration > Communication Settings (Power Supply)

3

The parameters in this group are used by inDART-HC08 to determine how to force the target microcontroller to enter the monitor mode. Depending on your target hardware, two possibilities are available:

- **Power control (automatic).** Select this option if you want inDART-HC08 to automatically turn off and on the target system in order to enter the monitor mode. In order for this option to work, the target system must be powered through the “VDD OUT” connector.
- **Power control (manual).** Select this option if your target system cannot be powered by the “VDD OUT” connector. You will be asked, when necessary, to power off and on the target board manually.

When “**Power control (automatic)**” is selected, you must additionally set two parameters:

- “**Power down time**”. The time it takes for the power voltage to drop below 100 mV.
- “**Power up time**”. The time it takes for target microcontroller to get ready to communicate after the V_{DD} voltage becomes high.

Additionally, the “**VDD**” parameter must be set to the actual operating voltage of the target microcontroller. This parameter is fundamental for proper operations, since it affects (among other things) the signal conditioning of MON08 lines.

Using Existing Projects with inDART-HC08

If your project has been targeted to an emulator/simulator other than inDART-HC08 and you wish to use inDART-HC08 as the debugger for your project, please do the following:

1. CodeWarrior is interfaced to the inDART-HC08 engine through a so-called “GDI interface”. From the CodeWarrior HC08 debugger interface, select **Component > Set Target** and choose “**GDI Target Interface**”.
2. A dialog box will appear asking you to locate the GDI dll file needed to interface with inDART-HC08. Select the **inDART_HC08.dll** file located into the **\Program Files\SofTec Microsystems\inDART-HC08** directory.
3. CodeWarrior will then recognize inDART-HC08 as the target interface for your project. The *MCU Configuration* dialog box will appear allowing you to select the derivative you are working with.
4. On the CodeWarrior HC08 debugger interface a new menu (**inDART-HC08**) will be created. From this menu, select **Load** and locate the object file your project is based on.

Notes and Tips

Stop Command Handling

The “**Halt**” debugging command will not work unless the IRQ interrupt is properly handled. In particular, the following precautions must be taken in the application’s source code.

1. Global interrupts must be enabled (use the “cli” instruction);
2. The IRQ interrupt must be enabled;
3. The IRQ interrupt vector must be handled;
4. The IRQ handling routine must include the following code:

```
irq_isr:   bil irq_isr           ; Waits for the IRQ signal to go high
           swi                  ; Jumps to monitor code
           rti
```

5. Under these conditions, the TGT_IRQ# line is reserved; when it is driven low, an “**Halt**” debugging command is automatically recognized.

Breakpoints and Swi Instruction

The HC08's on-chip debug module only handles one hardware breakpoint. However, you can force the program execution to stop at other specific locations by inserting the “**swi**” Assembly instruction on your source code.

3

Reading Peripheral Status

Care must be taken when reading some peripheral's status/data registers, since a reading operations 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.

Interrupt Execution during Steps

When issuing stepping instructions (Single Step, Step Over, etc.) and there are pending interrupts, inDART-HC08 will not step inside the interrupt handling routine, but the whole interrupt handling routine is executed. An exception is when you single step on an Assembly instruction which branches to itself: in this case, interrupts which may occur are not handled.

Peripheral Status during Steps

When single stepping on an Assembly instruction which branches to itself, peripheral status is frozen.

Internal RC Oscillator Calibration

Devices which feature an internal RC oscillator need to be calibrated through the OSCTRIM register. Every time you erase the target device (with the DataBlaze programming utility) or enter the debugging mode (with CodeWarrior or inSIDE) the OSCTRIM register value is lost. We strongly recommend to take note of the OSCTRIM register value before erasing the device or entering a debugging session.

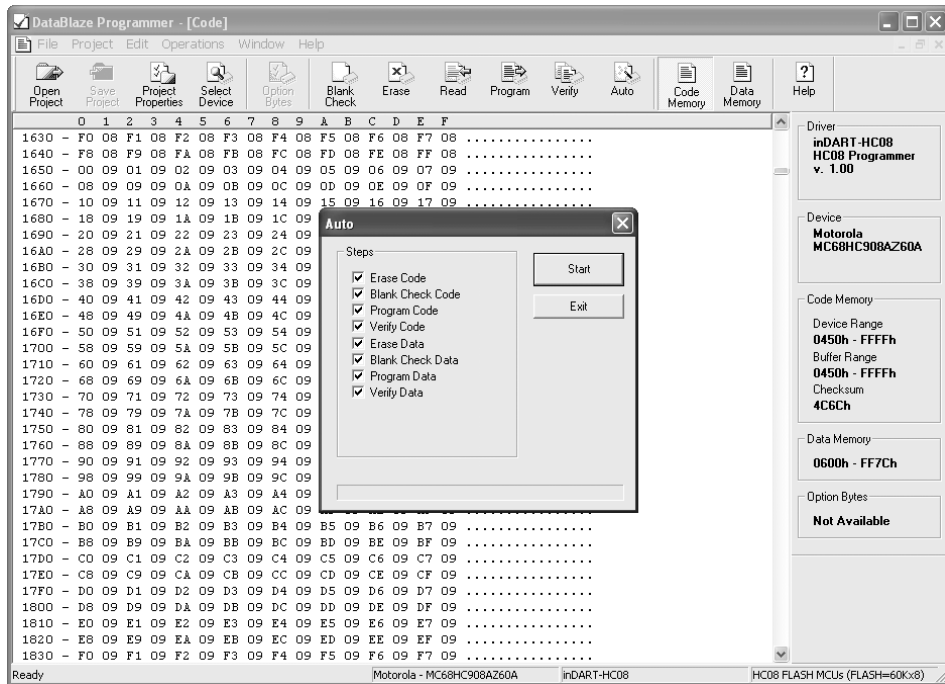
DataBlaze Programming Utility

A full-featured programming utility (DataBlaze) is also provided with inDART-HC08. To start the DataBlaze utility select **Start > Programs > SofTec Microsystems > inDART-HC08 > DataBlaze Programmer**.

DataBlaze offers the following advanced features:

- Code memory editing;
- Data memory (EEPROM) editing;
- Blank check/erase/program/verify operations on Code memory and Data memory;
- Read operations from Code memory and Data memory;
- Project handling;
- One-button, multiple-operations programming (“Auto” feature);
- Serial numbering.

3



The DataBlaze User Interface

SofTec Microsystems' inSIDE HC08 Debugger

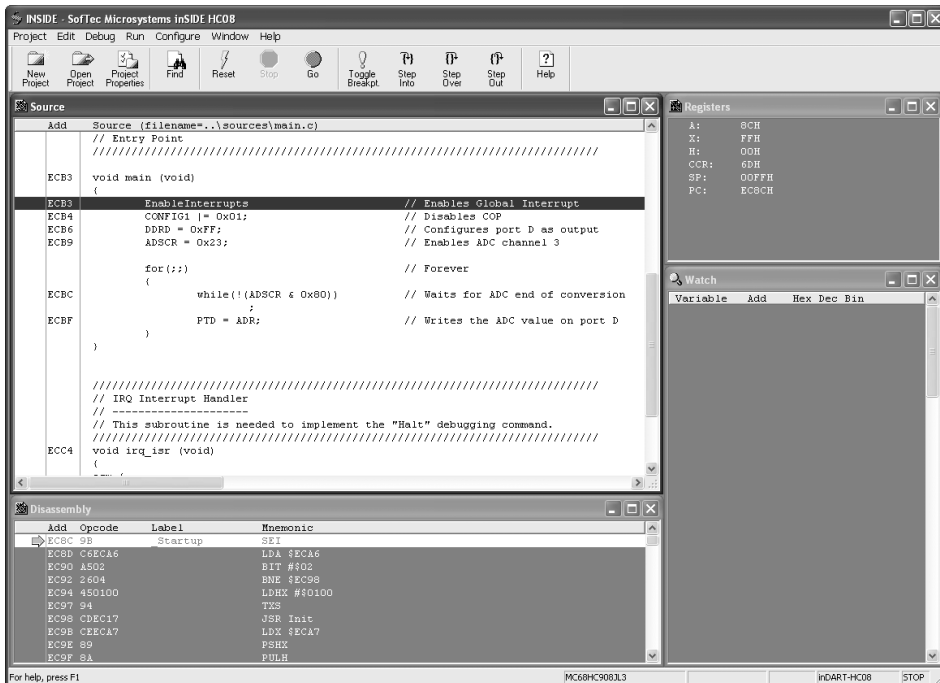
SofTec Microsystems' inSIDE HC08 debugger can be used as an alternative to the CodeWarrior HC08 IDE. SofTec Microsystems' inSIDE HC08 features a

3. inDART-HC08 Operations

basic, easy-to-use debugger, with no editor, assembler, or C compiler. However, the inSIDE HC08 debugger is not size-limited (it is not limited at all) and can additionally open ELF/DWARF, P&E MAP, Freescale S-Record, Intel-Hex and binary source files.

For detailed instructions on how to use inSIDE HC08, please refer to its online documentation.

3



The inSIDE HC08 User Interface

4. Troubleshooting

Common Problems and Solutions

This section reports some common problems that may arise during general use. However, working with a specific target device may cause device-specific issues.

Communication Can't Be Established with inDART-HC08

1. Make sure the inDART-HC08 in-circuit debugger is connected to the PC and powered on. inDART-HC08 is powered by the USB connection.
2. Make sure you are working with the correct inDART hardware model. To view/change the inDART hardware model in use, choose **inDART-HC08 > MCU Configuration** the CodeWarrior HC08 debugger's main menu.
3. If the **inDART-HC08** menu is not present in the CodeWarrior HC08 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, you must do the following:
 - From the **GDI** menu, choose **MCU Configuration** and verify that the hardware code and device code parameters are set correctly;
 - Press the "**Communication Settings**" button and verify that all of the communication parameters are set correctly.
4. Make sure the demo board/target application board is powered on and the target microcontroller is working. Programming and debugging rely on a MON08 serial communication between the inDART-HC08 board and the demo board/target application. This means that, in order to work correctly, the target microcontroller must be running. In particular, make sure that:
 - The MON08 cable is connected to the demo board/target application's MON08 connector.
 - The target microcontroller is in place.
 - All of the MON08 connector signals are correctly tied to the target microcontroller.

- The TGT_IRQ# and the RST_IN# MON08 lines aren't driven low by the target.
- The oscillator circuitry is working at the frequency required to generate the specified MON08 target's baud rate. To view/change the MON08 baud rate, choose **inDART-HC08 > MCU Configuration** from the CodeWarrior HC08 debugger's main menu and click the **"Communication Settings"** button.

5. Make sure that the target baud rate setting, required to enter the monitor mode, has been calculated taking into account the **"Frequency divider"** parameter (in the *Communication Settings* dialog box).

4

A Communication Error Is Returned on a Program Execution Command (Start/Continue, Single Step, etc.)

- Make sure that the target microcontroller's I/O bit corresponding to the MON08 line dedicated to the bidirectional communication is set to input by your program.
- Make sure that your program works correctly. Among other things, your program must not access reserved memory locations.

Peripheral Speed is Low

The **"Frequency divider"** parameter should always be set to 4 in order to guarantee that, in monitor mode, all of the target microcontroller's peripherals run at the same speed they do in user mode. If, however, you set this parameter to 2 not all peripherals will work at this doubled speed.

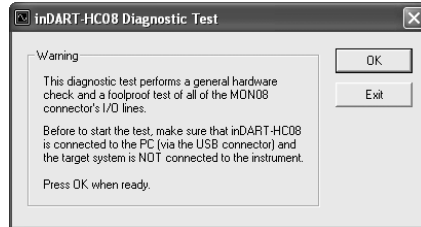
Stepping Execution is Slow

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

Diagnostic Test

inDART-HC08 has built-in self-test capabilities. This means that you can verify by yourself, at any time, the correct operation of the instrument's hardware. The diagnostic test is accessible through a small, separate test utility. To perform the diagnostic test:

1. Start the inDART-HC08 diagnostic test utility by selecting **Start > Programs > SofTec Microsystems > inDART-HC08 > inDART-HC08 Diagnostic Test**. The following dialog box will appear.



The *inDART-HC08 Diagnostic Test* Utility

2. Make sure that inDART-HC08 is connected to the host PC;
3. Make sure that no target system is connected to inDART-HC08;
4. Click “**OK**”. The test will be performed. In case of problems, please contact our technical support.

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.

To benefit the free technical support service, the Registration Card must have been filled out and sent in.

SofTec Microsystems offers its customers a free technical support service at support@softemicro.com. Before to get in contact with us we suggest you, however, to visit our online FAQ section and to be sure you are working with the latest version of the inDART-HC08 user interface (upgrades are available for free at <http://www.softemicro.com/download.html>).

Appendix A. Electrical and Physical Specifications

Operating Voltage	5 V DC (provided by the USB connection)
Power Consumption	300 mA (max)
V_{DD} (Voltage Supplied by the V_{DD} Line on the “VDD OUT” Connector)	2.2 V, 2.7 V, 3.3 V, 5.0 V
I_{DD} (Current Drawn from the Target via the V_{DD} Line on the “VDD OUT” Connector)	200 mA (max)
OSC Signal Characteristics (“OSC OUT” Connector)	6 or 12 MHz, GND to V _{DD} , 50% Duty Cycle
Dimensions	95 x 55 x 15 mm
Weight	25 g
Operating Temperature	0 °C to 50 °C
Storage Temperature	-20 °C to 70 °C
Humidity	90% (without condensation)

Electrical and Physical Specifications



Appendix B. MON08 Connector

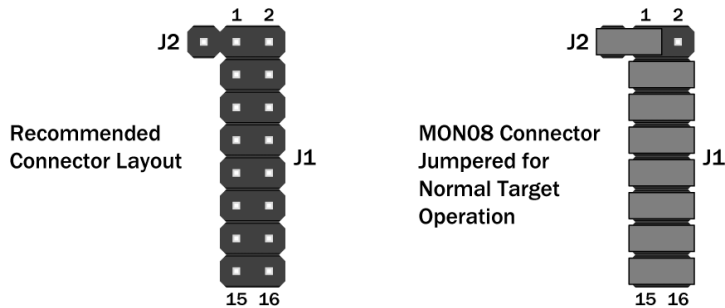
In-Circuit Debugging

It is recommended that your own application board features a MON08 interface realized with two connectors as illustrated below.

During debugging, the MON08 cable must be connected to the J1 connector (see figure below). The J2 connector is not used during debugging.

Standalone Target Operation

In order for the target board to work without being connected to inDART-HC08, the MON08 connector's pins must be jumpered as shown below.



MON08 Connector (Target Side)

B

