

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

RSKSH7201 USB Application Note

RSK Ethernet & USB Application Board

RENEASAS STARTER KITS

RENEASAS SINGLE-CHIP MICROCOMPUTER

SuperH™RISC engine

Table of Contents

Chapter 1. Preface	3
Chapter 2. Purpose	4
2.1. Software installation and connection details	4
2.2. Connection	5
2.3. Power – Up Behaviour	5
Chapter 3. Project Workspace	6
3.1. Introduction	6
3.2. Accessing the RSK_Apps Project Workspace	6
3.3. Build Configurations and Debug Sessions	7
3.3.1. Build Configuration	7
3.3.2. Debug Session.....	7
Chapter 4. Building the RSK_Apps Project.....	8
4.1. Building Code	8
4.2. Connecting to the target with the E8	9
4.2.1. Connecting to the target with the E8.....	12
Chapter 5. Downloading and running the demonstration	13
Chapter 6. The UsbTestBench Application	14
6.1. Installation of the USB device Drivers	14
6.2. Connecting to the device.....	16
6.3. USB Console	17
6.4. Data Tests.....	19
Chapter 7. USB Demonstration Firmware Architecture	20
7.1. USB Console	20
7.2. USB Host connection monitor	21
7.3. SCIF0 Console	21
7.4. LED Flash.....	22
7.5. USB Function Driver.....	22
7.6. USB Host Driver	22
Chapter 8. Additional Information.....	23

Chapter 1. Preface

Cautions

This document may be, wholly or partially, subject to change without notice.

All rights reserved. No one is permitted to reproduce or duplicate, in any form, a part or this entire document without the written permission of Renesas Technology Europe Limited.

Trademarks

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

Copyright

© Renesas Technology Europe Ltd. 2006. All rights reserved.

© Renesas Technology Corporation. 2006. All rights reserved.

© Renesas Solutions Corporation. 2006. All rights reserved.

Website: <http://www.renesas.com/>

Glossary

HMON	Embedded Monitor	RSK	Renesas Starter Kit
LCD	Liquid Crystal Display	CPU	Central Processing Unit
LED	Light Emitting Diode	USB	Universal Serial Bus
OTG	On The Go	HNP	Host Negotiation Protocol
VID	Vendor Identifier	PC	Personal Computer
PID	Product Identifier	RTOS	Real Time Operating System
IOID	Input Output Identifier	PTD	Phillips Transfer Descriptor
ATL	Acknowledged Transfer List		

Chapter 2. Purpose

This manual is designed to enable the user to setup and use the RSK Ethernet & USB Application Board for USB communications in conjunction with the standard Renesas Starter Kit for SH7201 (hereafter referred to as RSKSH7201). The manual also seeks to answer, in tutorial form, the most common questions asked about using the combination of RSKSH7201 and the RSK Ethernet & USB Application Board.

The RSK CD contains a tutorial project for use with the RSKSH7201 and the RSK Ethernet & USB Application Board.

Files referred to in this manual are installed using the project generator as you work through the tutorials. The tutorial examples in this manual assume that installation procedures described in the RSKSH7201 Quick Start Guide have been completed.

Please refer to the Quick Start Guide for details of installation and configuration of the RSKSH7201.

NOTE: These tutorials are designed to show you how to use the RSKSH7201 and are not intended as a comprehensive introduction to the High-performance Embedded Workshop (HEW) debugger, the compiler tool-chains or the E8 Emulator – please consult the relevant user manuals for more in-depth information.

2.1. Software installation and connection details

A Philips ISP1761 High Speed Universal Serial Bus On-The-Go controller is used to provide the USB functionality.

Features include:

- USB Device Controller sample application.
- Windows PC demonstration application.
- USB Host Controller sample application.
- Sample USB Device Controller driver code.
- Sample USB Host Controller driver code.

This manual describes the technical details of the demonstration firmware and use of the PC demonstration application. This demonstration does not cover OTG or HNP functions of the USB controller.

It is preferable to have attempted the basic RSKSH7201 tutorial prior to using the RSK Ethernet & USB Application Board. Doing so will ensure that all the software and USB drivers are installed correctly. For this, please refer to the RSKSH7201 Quick Start Guide. Once completed, please return to this point.

All the software to support the RSK Ethernet & USB Application Board will have been installed by using the default settings on the installer. Now ensure that the RSKSH7201 is not connected or powered and remove the LCD panel, this is not required for this tutorial.

The RSK Ethernet & USB Application Board is shipped with connectors for the user to apply to the RSKSH7201, the tools required are a soldering iron and suitable lead-free solder (leaded solder maybe used but this would invalidate the lead-free status of the RSKSH7201 and the RSK Ethernet & USB Application Board).

1. The two 24 way sockets should be soldered on the **underside** of the target RSKSH7201 in JA5 and JA6 positions.
2. The two 26 way sockets should be soldered on the **underside** of the target RSKSH7201 in JA1 and JA2 positions.
3. The 50 way socket should be soldered on the **underside** of the target RSKSH7201 in the JA3 position.

Hint: An easy way to do this is to leave the connectors on the RSK Ethernet & USB Application Board and place the RSKSH7201 on the top, then solder the connectors with the boards in position.

2.2. Connection

Please refer to the photograph:

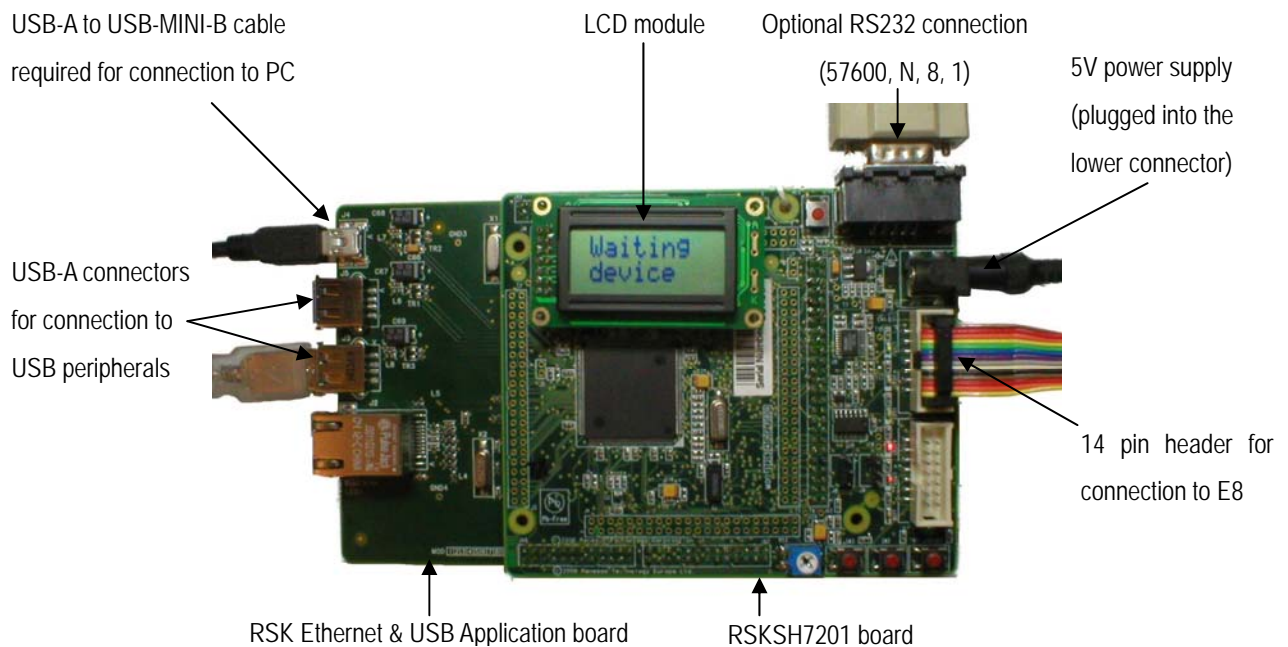


Fig 2.1 Connections to the system

1. Connect the RSK Ethernet & USB Application Board to the RSKSH7201 ensuring careful placement of the pins into the sockets.
2. Connect the E8 debugger module to the RSKSH7201. Also connect the USB cable between the E8 and the host PC.
3. Insert the centre positive, 5V power lead to the connector on the RSKSH7201 – NOT the one on the RSK Ethernet & USB Application Board, but do not switch the power-on yet..

2.3. Power – Up Behaviour

When the RSKSH7201 is purchased, the CPU board has the 'Release' or stand alone code from the demonstration code pre-programmed into the Renesas microcontroller. On powering up the RSKSH7201 board, the user LEDs will start to flash.

Chapter 3. Project Workspace

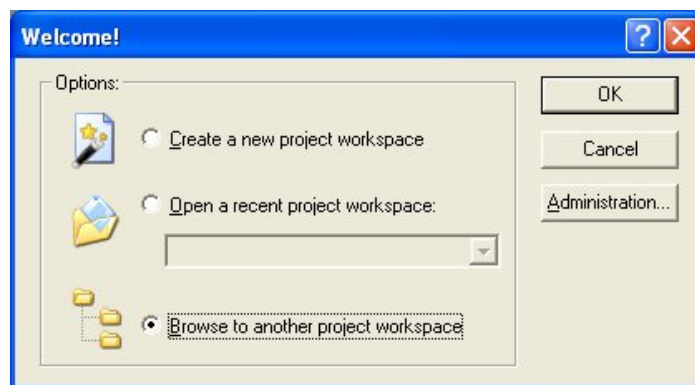
3.1. Introduction

High-performance Embedded Workshop (HEW) is an integrated development tool that allows the user to write, compile, program and debug a software project on any of the Renesas Microcontrollers. HEW will have been installed during the installation of the software support for the RSK product.

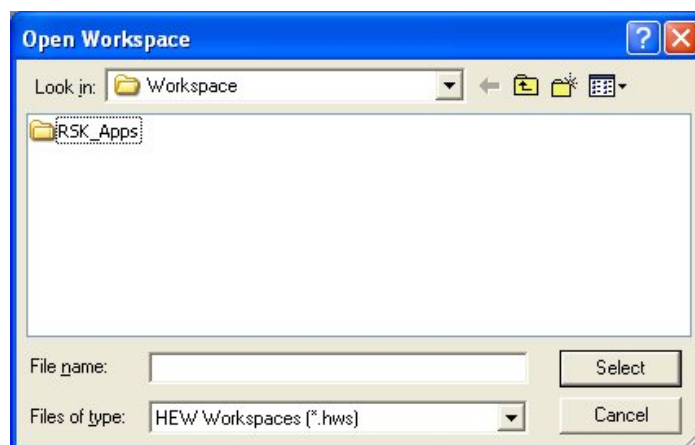
3.2. Accessing the RSK_Apps Project Workspace

Unlike standard RSK boards which are supplied with HEW Project generators, the RSK Ethernet & USB Application Board is supported by a pre-made HEW Workspace; this supports both the Ethernet and USB modes of operation and will have been placed in the c:\Workspaces folder during the installation procedure described in Chapter 2. The workspace requires only to be built and the communications configured.

1. Launch HEW
2. The HEW Welcome! dialog will be shown:



3. Select "Browse to Another Workspace", click <OK>
4. On the "Open Workspace" view the default is to look in the "Workspace" folder, you will find the "RSK_Apps" workspace folder there.



5. Double click on the "RSK_Apps" folder.
6. Double click on the "Comms" folder.

7. Double click the "USB" folder and then select the "RSKSH7201.hws" workspace file within, click <Select>
8. The workspace will open into the HEW showing the project files.

3.3. Build Configurations and Debug Sessions

The workspace that has been created contains two build configurations and two debug sessions. The Build Configuration allows the same project to be built but with different compiler options. The options available to the user are fully described in the HEW User's Manual.

3.3.1. Build Configuration

The build configurations are selected from the left hand drop down list on the tool bar. The options available are Debug and Release. The debug build is configured for use with the debugger. The Release build is configured for final ROM-able code.

A common difference between the two builds may be the optimisation settings. With Optimisation turned on the Debugger may seem to execute code in an unexpected order. To assist in debugging it is often helpful to turn optimisation off on the code being debugged.

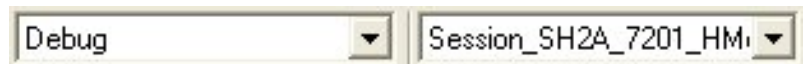
- Select the 'Debug Build' Configuration.



3.3.2. Debug Session

The debug sessions are selected from the right hand drop down list on the tool bar. The options vary between RSK types however one will always start Debug and include the type of debug interface, the alternate selection is the 'Default' Session. The purpose of the debug sessions is to allow the use of different debugger tools or different debugger settings on the same project.

- Select the 'Session_SH2A_7201_HM' debug session.



Chapter 4. Building the RSK_Apps Project

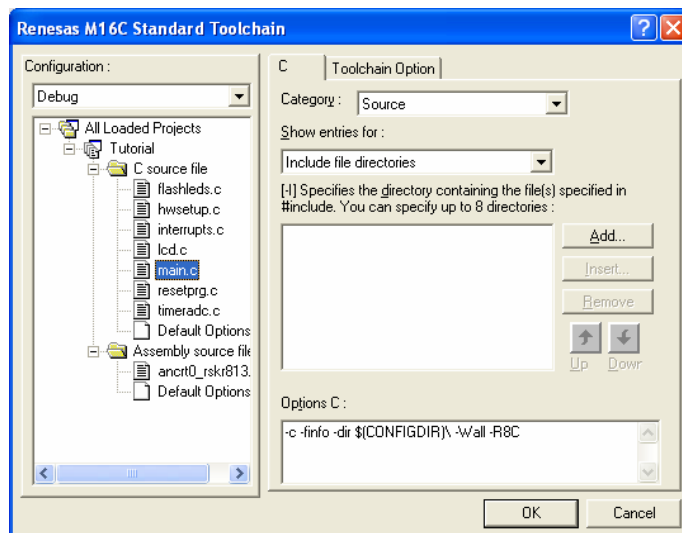
The project build settings have been pre-configured in the tool-chain options. To view the tool chain options select the 'Build' Menu item and the relevant tool-chain. This should be the first option(s) on the drop down menu.

The dialog that is displayed will be specific to the tool-chain selected.

The configuration pane on the left hand side will exist on all the tool-chain options. It is important when changing any setting to be aware of the current configuration that is being modified. If you wish to modify multiple or all build configurations this is possible by selecting 'All' or 'Multiple' from the 'Configuration' drop down list.

- Review the options on each of the tabs and 'Category' drop down lists to be aware of the options available.



Note: For the purposes of the tutorial, leave all options at default.



When complete close the dialog box by clicking <OK>.

4.1. Building Code

There are three short cuts available for building the project.


- Select the 'Build All' tool bar button. 
This will build everything in the project that has not been excluded from the build. This includes the standard library and the two dependant libraries (Platform and CPU).
- Select the 'Build' tool bar button. 
This will build all files that have changed since the last build. The standard library will not be built unless an option has been changed. If the source code in either of the two dependant libraries has been changed they will be built too.
- Press 'F7'
This is equivalent to pressing the 'Build' button described above.
- Build the project now by pressing 'F7' or pressing one of the build icons as shown above.

During the build each stage will be reported in the Output Window.

The build will complete with an indication of errors and warnings encountered during the build.

4.2. Connecting to the target with the E8

This section will take you through the process of connecting to the device, programming the Flash and executing the code.

- Select the "Configure Flash project" button from the toolbar 

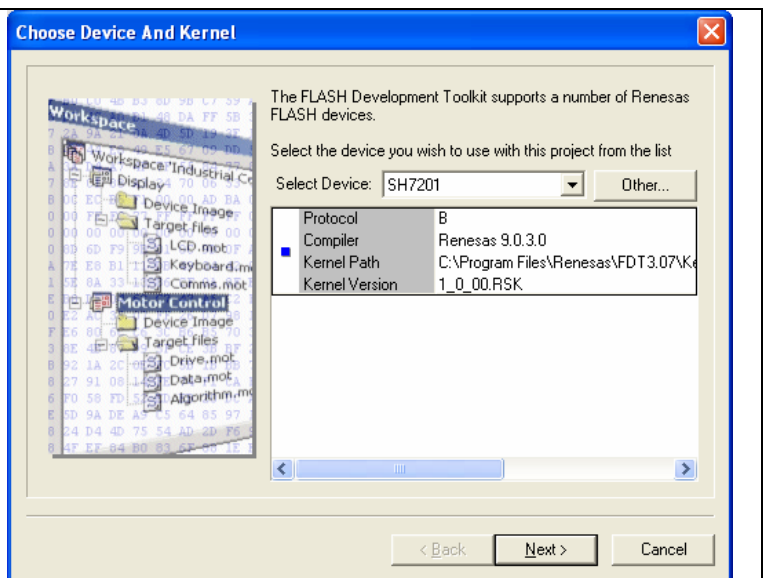
If the flash kernel is already configured then a confirmation window will open with the kernel settings. To modify any of the settings just double click on the item. If FDT is already configured then please proceed to section 4.2.1

The first stage is to select the specific device from the installed kernels. The installer will have added the kernel to the FDT registry so it will appear under the device list drop down menu. If for any reason the kernel has to be re-build such as the crystal frequency on the board has changed, then the installation path is: <FDT Installation>\Kernels\ProtB\7201\Renesas\1_0_00.RSK

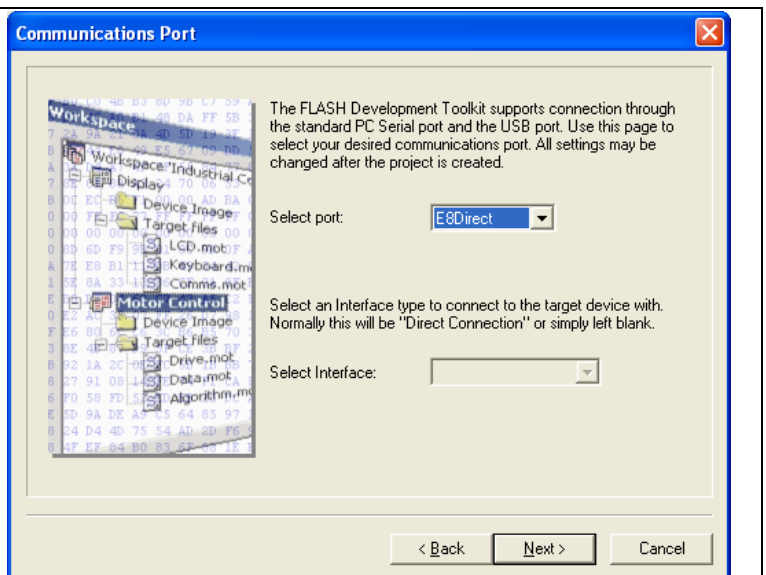
- Select the Device on the RSK from the drop down list.
- In the sub pane, select the kernel version that ends in '.RSK'.
- Press <Next>.

If you have copied the kernel to another location to modify for a different crystal frequency, the device will not be listed in the sub pane.

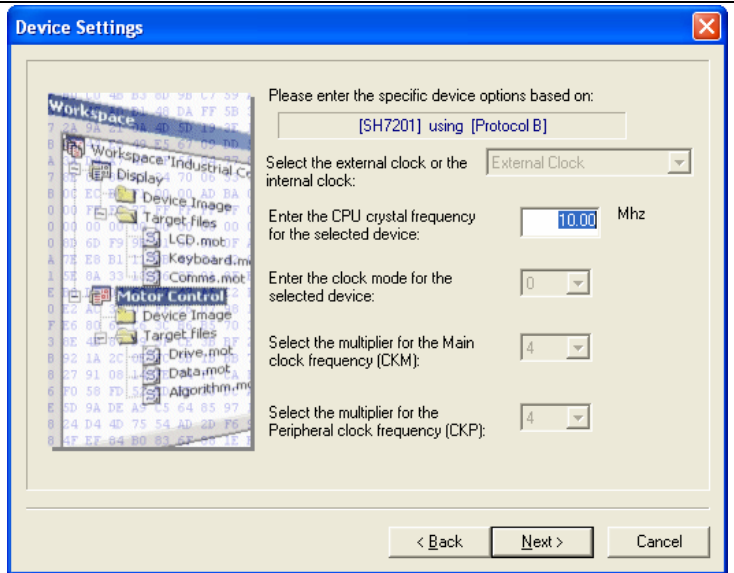
- Press <Other...> and navigate to your modified kernel.



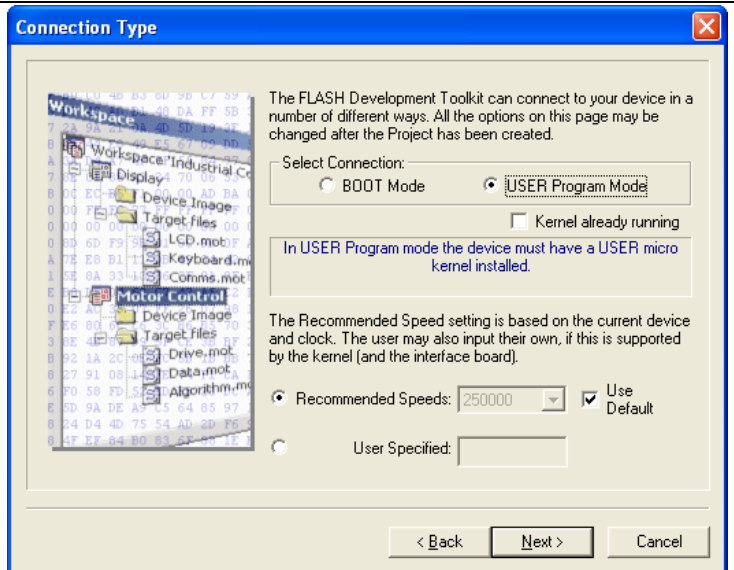
- Select E8DIRECT as the communication Port
- Press <Next>.



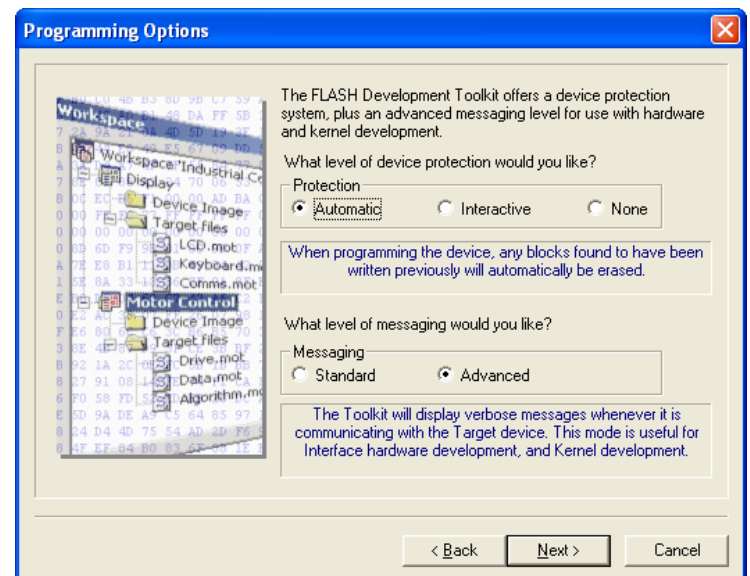
- The default settings are suitable for an un-modified RSK board.
- Confirm the Crystal Frequency matches the board.
- Confirm the main clock frequency multiplier.
- Confirm the peripheral clock multiplier.
- Press <Next>.



- Ensure that 'USER Program Mode' is selected.
- Confirm that 'Use Default' is selected.
- Press <Next>.

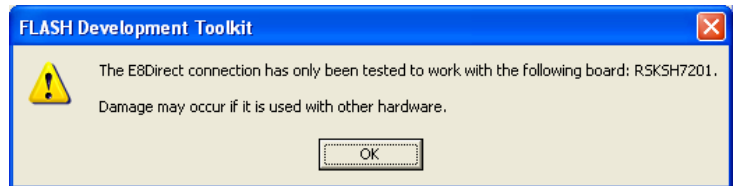


- Confirm the default selections of 'Automatic' and 'Advanced'.
- Press <Next>.



The following warning dialog will be displayed.

- Press <OK>

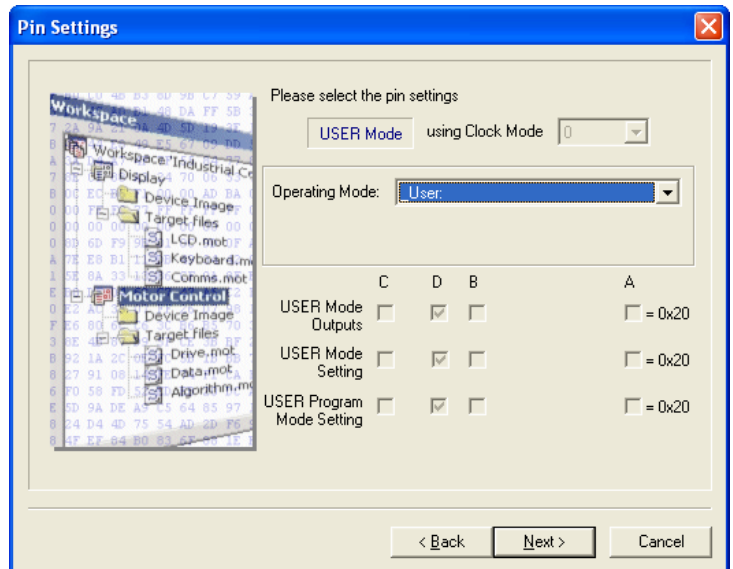


To communicate with the RSK, FDT and HMon need to be able to change the operating mode of the microcontroller. To do this there are settings to control the state of the Mode pins via the E8Direct interface. These settings are confirmed in the following screens.

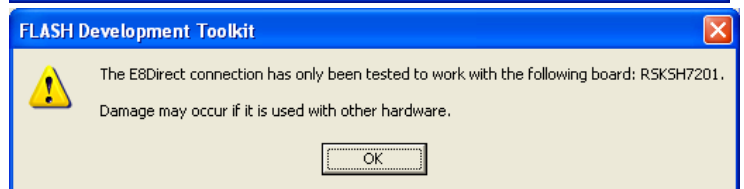
Unless you are very sure that the mode pin settings need to change. Do not modify the default settings.

Damage to the microcontroller can be sustained with incorrect settings.

- Confirm the mode pin settings.
- Press <Next>.



- The following warning dialog will be displayed.
- Press <OK>.

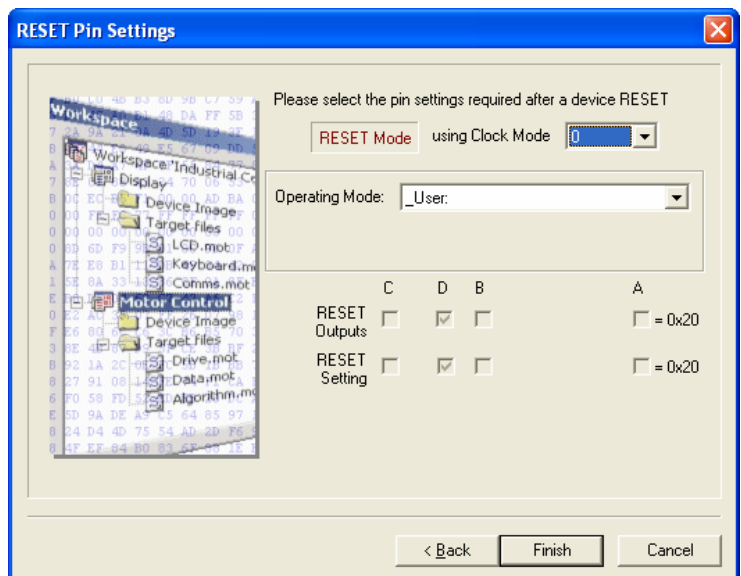


Note: Damage to the microcontroller can be sustained with incorrect settings.



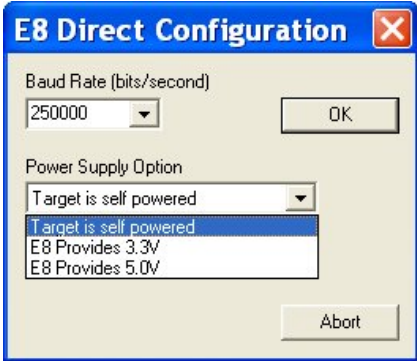
- Confirm the mode pin settings.
- Press <Finish>.
- The Flash configuration has now been completed.

If you have changed any workspace settings now is a good time to save the workspace.

- Select [File' -> 'Save Workspace'].



4.2.1. Connecting to the target with the E8

<ul style="list-style-type: none"> • Provide power to the hardware – refer to chapter 2 for connection details 	
<ul style="list-style-type: none"> • Click the <Connect> button on the debug toolbar 	
<p>HMon is able to discover the internal flash configuration of the device from the FDT kernel we configured earlier. HMon also needs information on the location of the IO registers and internal / external RAM. This information is stored in a '.TCF' file. The TCF file is supplied and registered with HEW. As it is possible to have TCF files with slightly different configurations the following dialog is displayed to allow selection.</p>	
<ul style="list-style-type: none"> • When the "Select USB Device" dialog appears, click <OK>. <p>Notes: If the E8 is not set up as "E8DIRECT" then please follow the on-screen instructions and refer to the RSKSH7201 Tutorial Manual.</p> <p>If running the E8 for the first time, the "Found Hardware Wizard" may automatically run after a short delay and install the appropriate drivers.</p>	
<ul style="list-style-type: none"> • The next dialog is the E8 Direct Configuration, select "Target is self powered" and click <OK>. • If the target will not connect it may be because HMON is not resident, click <Abort> then select "Download HMON using Boot Mode", <OK> then re-attempt the E8 Direct connection. <p>Now is a good time to save the HEW session.</p> <ul style="list-style-type: none"> • Select 'File' 'Save Session'. <p>If you have changed any workspace settings, save the workspace.</p> <ul style="list-style-type: none"> • Select 'File' 'Save Workspace'. 	

Chapter 5. Downloading and running the demonstration

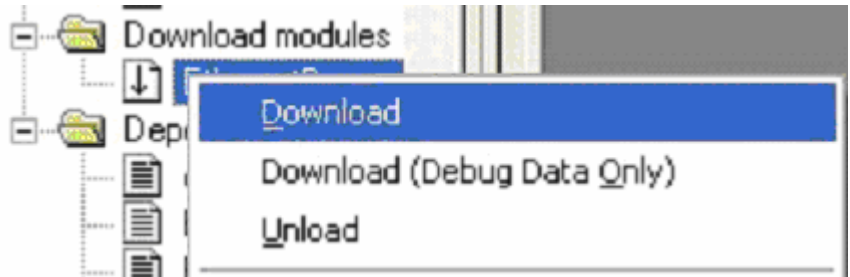
NOTE: The USB communications should be disconnected from the PC before code is downloaded

Once the code has been built in HEW it needs to be downloaded to the RSK itself.

There will now be an additional category in the workspace view for 'Download Modules'

- Right click on the download module listed and select 'Download'

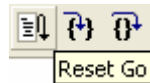
Note: If shown the "HMON Configuration" dialog, select "Compare with Cache" and click <OK>.



On completion of the download the debugger is ready for the code to be executed.

To start the sample code it is necessary to reset the debugger and target and start the code, this can be done in one hit.

- Press 'Reset Go' on the Debug Tool Bar.



The USB demonstration application is now running under the control of the debugger.

Chapter 6. The UsbTestBench Application

The UsbTestBench application and USB device drivers will have been installed during the software installation for the RSK product. When the device is connected to the PC for the first time the USB device drivers will need to be installed. After driver installation has completed the UsbTestBench application can be run.

1. Ensure that the USB Demo code is downloaded successfully and it is running on the target.
2. Connect the "USB-A to USB-Mini-B" cable between host PC and J4 connector of the RSK Ethernet & USB Application Board.
3. Wait for a while for the PC to detect the new USB device first time.

6.1. Installation of the USB device Drivers

The device driver installation process varies between different versions of Windows operating systems. As a general rule follow the prompts "recommended" by the dialogs. It is also important to note that with the professional versions of Windows operating systems administrators rights are required to install a device driver.

When the device is plugged in the Found New Hardware Wizard dialog is displayed:

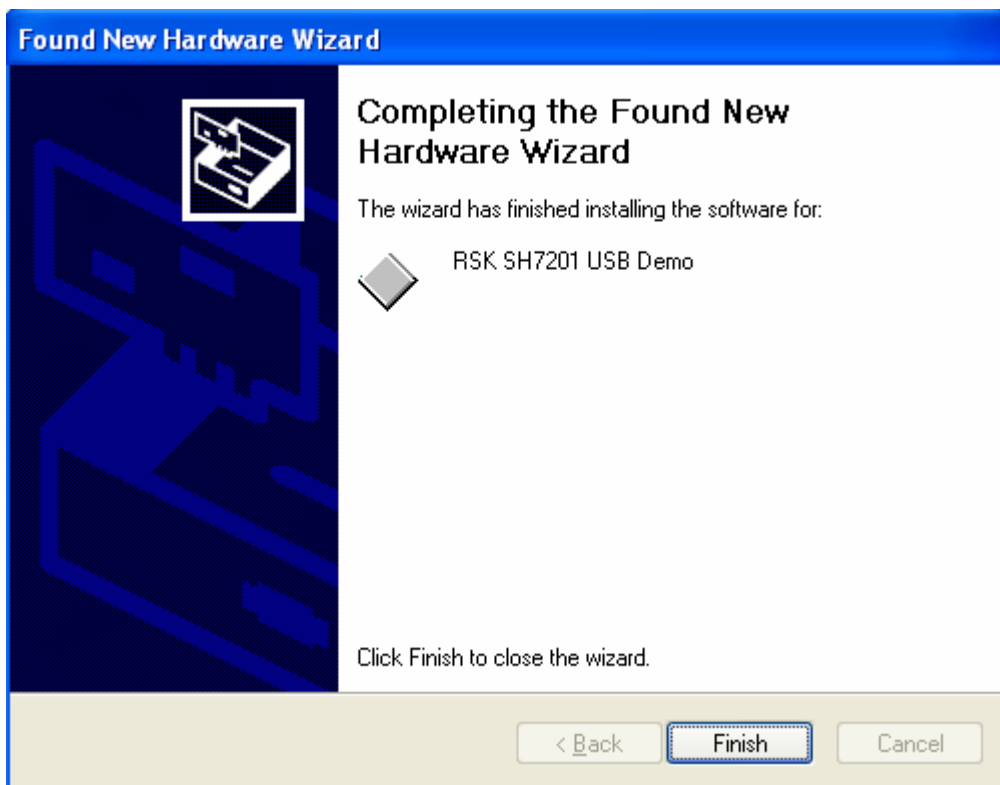


Select the Install the software automatically (Recommended) radio dial and click the next button.

The following dialog will be displayed as the driver files are installed:



Once the driver is installed click the Finish button on the dialog below to complete the driver installation.



6.2. Connecting to the device

The UsbTestBench application can be launched from the start menu:



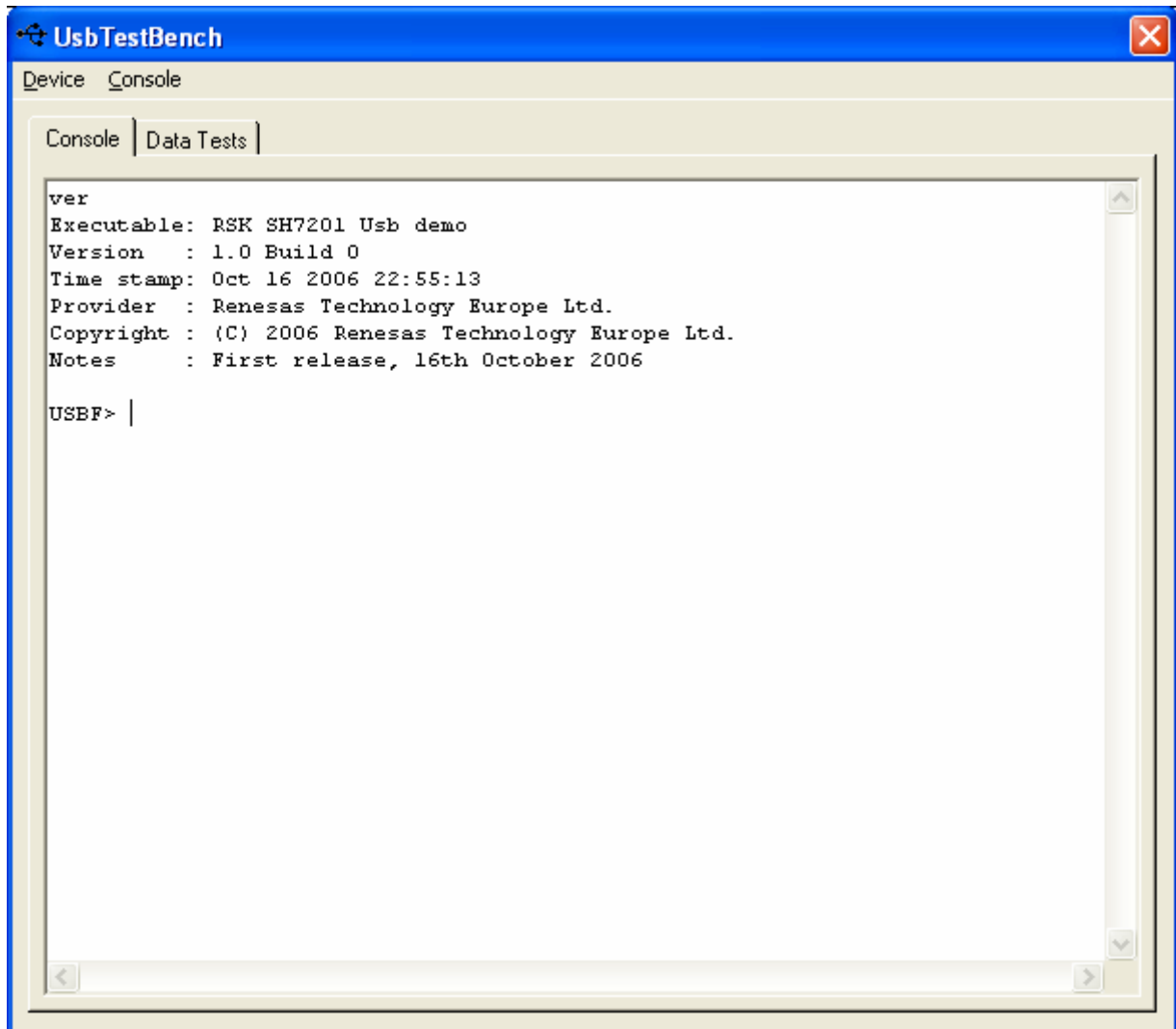
When the application opens the select device dialog will show the available devices. If there are no devices available the list box will be empty. In this case click Cancel, plug in the device and then select Open from the Device menu.



Click OK to connect to the device.

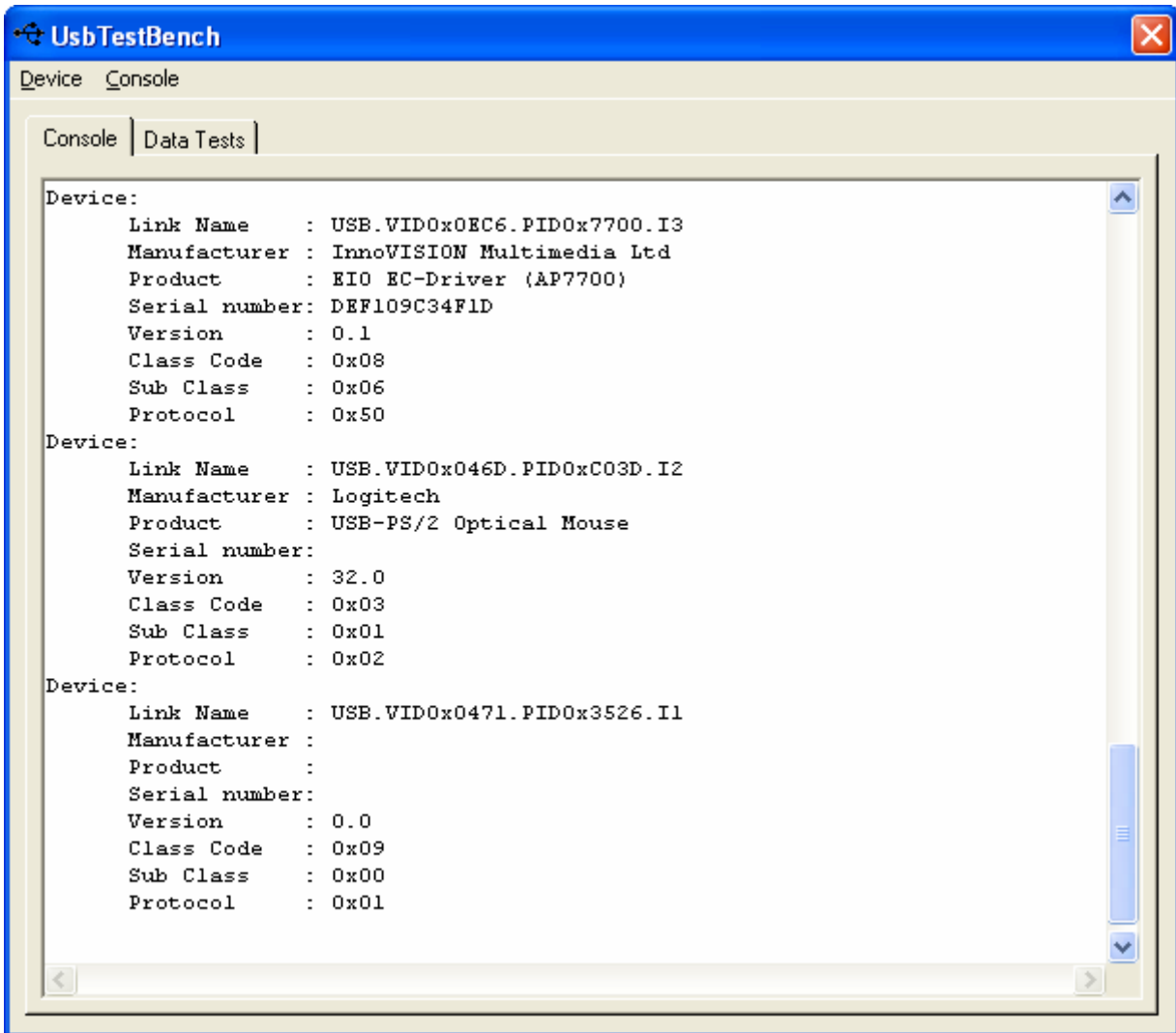
6.3. USB Console

Once the UsbTestBench has connected to the device the application opens the command console.



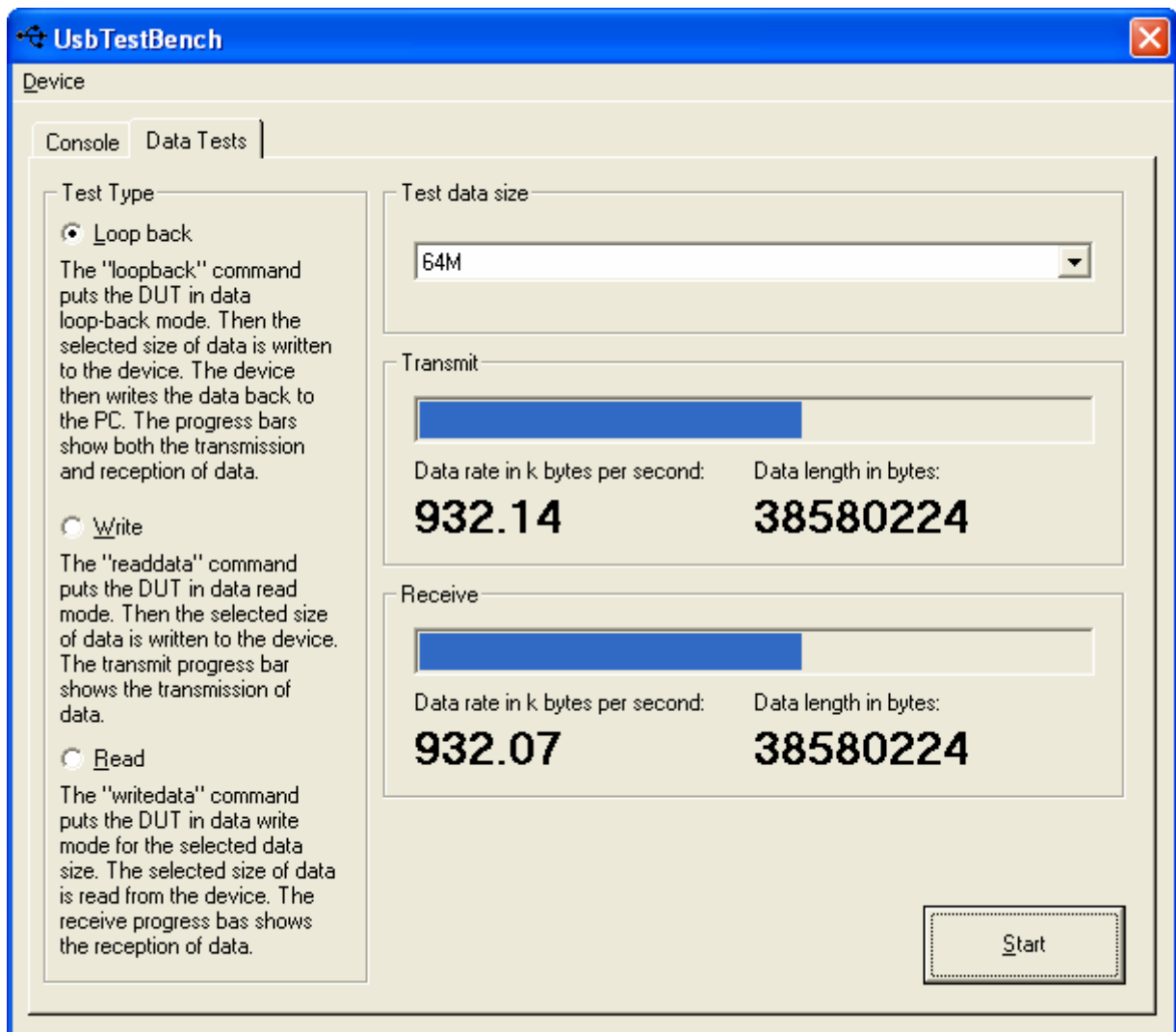
The console window should contain the firmware version information and a command prompt. The commands "?<CR>" and "??<CR>" list the commands available.

The "device" command lists the devices connected to the RSK Ethernet & USB Application Board. If a USB peripheral is now connected to one of the USB-A ports it can be seen in the device list. The last device is the root hub which is internal to the ISP1761. The screen shot below shows the device list when a high speed USB hard-disk and low speed USB-PS/2 optical mouse are connected to the USB-A ports.



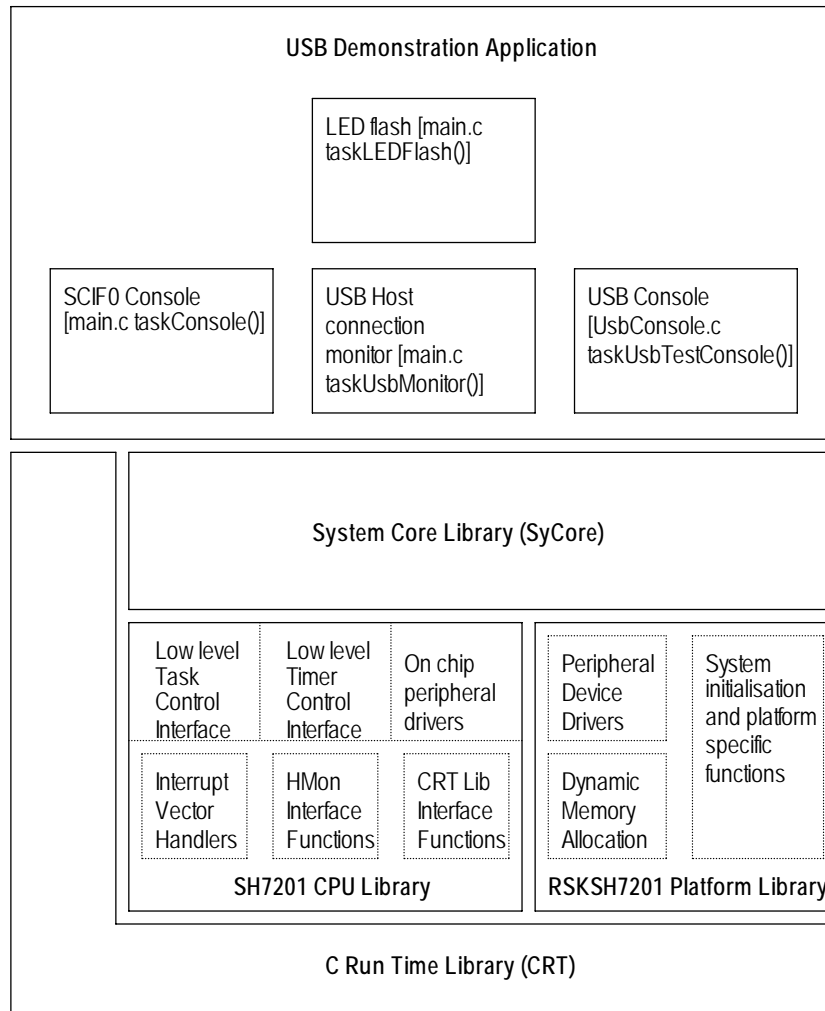
6.4. Data Tests

To quantify the performance of the system, the Data Test tab provides three simple tests. These are loop back, write and read. In these tests data is transferred between the PC and the device showing rate and progress of the transfer. Select the desired transfer size from the drop down list box and select the required test type radio dial. Click the start button to start the test. The console tab is inoperative until the test has finished. The transfer rate will vary between PC's and the connection topology of the device. There are also various inaccuracies in the transfer rate measurement method. These inaccuracies become smaller with larger transfer sizes.



Chapter 7. USB Demonstration Firmware Architecture

Due to the complexity of the USB Host and Function software this hardware test / demonstration code uses a library (SyCore.lib) which provides an RTOS. The RTOS allows the code to be broken up into tasks, simplifying the design process. The diagram below shows how the demonstration application is divided.



The Project Workspace reflects this in that it contains the two library projects called CPU (SH7201 CPU library) and Platform (RSKSH7201 Platform library). The CPU library contains drivers and functions specific to the SH7201 RISC Microcomputer. The Platform library contains drivers and function specific to the RSKSH7201 and the RSK Ethernet & USB Application Board. The C run time library and System Core library are not supplied in source code form so they do not have associated Project Workspaces.

7.1. USB Console

The USB console task performs the test functions for the UsbTestBench PC application. It processes the commands and performs the loop back, read and write data transfer tests. The source code for this is in the file UsbConsole.c. The functions communicate with the USB Function device driver (Device Controller) through an IOID. This abstracts the application from the implementation of the USB Function driver, allowing it to use the standard library functions IORead and IOWrite to exchange data with the PC. Methods specific to the USB Function driver are implemented via the standard library function IOControl. For more details on IORead, IOWrite and IOControl see the SyCore reference manual. The USB Function driver responds to four special control codes, detailed in the table below.

IOControl Code	Function
IOCTL_STD_HOST_CONNECTED	Reports on the host connection status
IOCTL_USBF_SEND_ZERO_LENGTH	Sends a packet with zero data length
IOCTL_USBF_SET_CONCAT	Prevents the driver terminating the transfer with a zero data length packet
IOCTL_USBF_SET_STALL	Sets the endpoint stall flag

The first standard control code is fairly obvious. When the PC has enumerated the device the IOCTL_STD_HOST_CONNECTED IOControl will report that the host is connected. When the cable is disconnected from the PC it will report that it is not connected. This enables the firmware to detect when the connection to the host has been lost and take any action required to allow re-connection to take place. The other three require a little more knowledge of the USB protocols to understand. Chapter 8 of the USB 2.0 specifications (usb_20.pdf) covers this in tedious detail. For the understanding of the above controls it is sufficient to say that a USB bulk data transfer is completed when either the requested length of data is transferred or a short packet is received. A short packet is any data transfer which contains less data than the endpoint packet size. For example, if the endpoint packet size is 512 bytes and the host has a 1024 byte transfer scheduled, when the first packet is received the transfer is not terminated. If the second packet is less than 512 bytes the transfer is retired before the 1024 bytes have been received. If there was only 512 bytes to transfer, then a packet with no data in it (zero data length) would need to be sent to retire the transfer. This function is supported by the IOCTL_USBF_SEND_ZERO_LENGTH IOControl. So as to provide the application with a simple interface, the driver will ensure that each call to IOWrite will complete any scheduled transfer. However, there are cases when the application may not want the driver to do this. The IOCTL_USBF_SET_CONCAT IOControl prevents the driver terminating the transfer with a zero length packet. It must be called before the call to IOWrite. This control will not work if the data transfer is not a multiple of the endpoint packet size. The endpoint packet size can be either 64 bytes (Full) or 512 bytes (High), depending on the speed of the hub the device is connected to. The application would need to know what the endpoint packet size was to take advantage of this functionality.

7.2. USB Host connection monitor

This task monitors a queue from the Host Controller Driver's enumeration task. When the enumeration task successfully enumerates a device it queues the product string (from the product string descriptor) or the symbolic link name of the device (containing the VID and PID). This string is read by the connection monitor task and scrolled across the LCD. A queue is a simple method that allows tasks to communicate with each other. The queue is created by IOOpen in the main function. The IOID of the queue is passed to the Host Controller Driver through the configuration pointer when it is opened. The IOID of the queue is also passed to the connection monitor task when it is created.

7.3. SCIF0 Console

This is similar to the USB console but uses SCIF0. It does not have the loop back, read and write functions of the USB Console. This task can be used to shut the system down and place the SH7201 in sleep mode via the "exit<CR>" command. In the case of a battery powered application the CPU may need to be placed into a low power mode to reduce power consumption. The Serial port is configured for a Baud rate of 57600, with 8 data bits, 1 stop bit and no parity.

7.4. LED Flash

This is a very simple task that flashes the LEDs on the RSKSH7201board to show the user that the application code is running.

7.5. USB Function Driver

The USB Function driver (Device Controller) can be found in the platform library. It consists of four files, namely DDUSBF.c which is the main driver file. DDUSBFci.c provides controller interfaces for reading and writing to the ISP1761 device controller and Ch9.c provides the protocol handling required by Chapter 9 of the USB specifications. The file USBCfg.c contains the device and configuration descriptors required by Chapter 9 of the USB specifications.

The ISP1761 has 7 user configurable endpoints. This driver implementation only uses one endpoint which is configured as type BULK. Endpoint 0 is the default control pipe and must be supported by all USB devices. It is this endpoint which allows the USB host to enumerate the device onto the bus. A full function driver implementation would take a configuration supplied by the application through the IOOpen function and configure the endpoints as described by the configuration data.

7.6. USB Host Driver

The USB host driver (Host Controller Driver) only enumerates devices to test the hardware. Hubs (other than the root hub) are not supported by this driver. The driver can be found in the platform library and consists of five files. DDUSBH.c is main driver file, host controller and root hub initialisation and control routines can be found in this file. The file DDUSBHci.c contains the controller interface functions. DDUSBHTL.c contains simple functions for maintaining the scheduled transfer lists. It is in this file that the function which writes the PTDs can be found. Only the structure of an ATL PTD has been defined in DDUSBHci.h. The main enumeration task can be found in the file DDUSBHCTL.c.

The ISP1761 and the SH7201's bus interfaces are initialised in the file DDISP1761.c. Both the host and function drivers are 'children' of the ISP1761 driver.

Chapter 8. Additional Information

For details on how to use High-performance Embedded Workshop (HEW), refer to the HEW manual available on the CD or installed in the Manual Navigator.

For information about USB 2.0 refer to the *USB 2.0 specifications*.

For more information about the USB Function driver see the *ISP1581 programming guide* and the *ISP1761 product data sheet*.

For more information about the USB Host driver see the *Enhanced Host Controller Interface Specifications* and the *ISP1761 product data sheet*.

For information about the SH7201 microcontrollers, refer to the *SH7201 Group Hardware Manual*.

For information about the SH7201 assembly language, refer to *SH-2A, SH2A-FPU Software Manual*.

Further information available for this product can be found on the Renesas website at:

http://www.renesas.com/renesas_starter_kits

General information on Renesas Microcontrollers can be found at the following websites.

Global: <http://www.renesas.com/>

Regional (English language) sites can be accessed from the Global site, or directly by going to:

Europe: <http://eu.renesas.com>

Americas: <http://america.renesas.com>

Asia: <http://sq.renesas.com>

RSKSH7201 USB Application Note

RSK Ethernet & USB Application Board

Publication Date Rev.01.00 15.JAN.2007

Published by: Renesas Technology Europe Ltd.

©2006 Renesas Technology Europe and Renesas Solutions Corp., All Rights Reserved.

RSKSH7201 USB Application Note RSK Ethernet & USB Application Board



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REG05J0002-0100