PowerPC cross development with GNU Toolchain and Eclipse

Version 1.0



Ronetix GmbH Waidhausenstrasse 13/5 1140 Vienna Austria Tel: +43-720-500315 +43-1962-720 500315 Fax: +43-1- 8174 955 3464 Internet: www.ronetix.at E-Mail info@ronetix.at

Acknowledgments:

ARM, ARM7, ARM9, and Thumb are trademarks of ARM Ltd. PowerPC is trademark of IBM. Windows, Win32, Windows CE are trademarks of Microsoft Corporation. Ethernet is a trademark of XEROX. All other trademarks are trademarks of their respective companies.

© 2005-2008 RONETIX All rights reserved.

Change log

November 2006 - First release

1		.5
2	PEEDI JTAG EMULATOR INSTALLATION	.6
3	TOOLSET INSTALLATION ON LINUX	.7
4	TOOLSET INSTALLATION ON WINDOWS	11
5	WORKING WITH ECLIPSE	12
5.1	Adding a project	12
5.2	Configuring and working with the Eclipse built-in debugger	17
5.3	Using Insight debugger as an Eclipse external tool	23

1 Introduction

This User Manual will show you how to install the GNU Toolchain and Eclipse, how to compile and debug a simple example using the Freescale Evaluation board MPC5554EVB and PEEDI JTAG Emulator and Flash Programmer.

The necessary software components for a PowerPC cross development are:

- GNU toolchain (compiler, linker, gdb)
- Eclipse IDE including Zylin CDT Plug-in for C/C++ development
- Java Runtime

To enable a quick start in the PowerPC cross development Ronetix provides pre-built packages for Linux and Windows hosts.

The necessary files for a Linux host are:

- The GNU toolchain: <u>http://download.ronetix.info/toolchains/powerpc/ronetix-powerpc-eabi-4.3.1.tar.bz2</u>
- The Eclipse IDE: <u>http://download.ronetix.info/eclipse/eclipse.tar.bz2</u>
- A simple example: <u>http://download.ronetix.info/examples/mpc5554_example.tar.gz</u>

The necessary file for a Windows host is:

http://download.ronetix.info/toolchains/powerpc/ronetix-toolset-ppc.exe

This is a Windows installer which installs the GNU toolchain, Eclipse IDE + Zylin CDT and Java.

2 PEEDI JTAG Emulator Installation

PEEDI (Powerful Embedded Ethernet Debug Interface) is an EmbeddedICE solution that enables you to debug software running on PowerPC processor cores via the JTAG port.

In order to debug you need to configure PEEDI JTAG Emulator. The configuration of PEEDI is common for both toolchains for Linux and Windows.

You can find detailed information about PEEDI in the PEEDI's User Manual: http://download.ronetix.info/peedi/doc/peedi_rev.A_manual.pdf

- Connect PEEDI to a free port of your LAN switch/hub using the supplied UTP patch cable.
- Connect PEEDI to the target using a JTAG cable and if needs the one of the supplied JTAG adapters. The JTAG adapter must be on the PEEDI side of the JTAG cable:



- Connect PEEDI to a COM port of your PC using the RS232 cable. Start any kind of terminal emulation program (HyperTerminal) and set it to 115200 bauds, 8 data bits, no parity and no flow control.
- Restart PEEDI holding pressed both front panel buttons to enter RedBoot command line.
- Use fconfig command to set the network configuration and other parameters.

WARNING:



If PEEDI is set to get its network settings from a DHCP server and if the Ethernet cable is unplugged or there is no DHCP server on the Ethernet, it may take some time for PEEDI to boot. To avoid this, make sure PEEDI can reach a DHCP server or assign a static IP address.

- Restart PEEDI again for the changes to take effect

After PEEDI is up and running (this should take some seconds after reset), press and hold the green front panel button and PEEDI will start to display its IP address on the display.

Connect to PEEDI with telnet application using the IP address from the previous statement. If connected, you should see the PEEDI CLI prompt

3 Toolset installation on Linux

To install the pre-built from Ronetix GNU cross-development tools:

Download the GNU tools form here:

http://download.ronetix.info/toolchains/powerpc/ronetix-powerpc-eabi-4.3.1.tar.bz2

or get it from the CD:

1. Mount the CD

mount /dev/cdrom /mnt/cdrom

2. Uncompress the archive

Cd / tar xvfj /mnt/cdrom/toolchain/ronetix-gnutools-powerpc-eabi.tar.bz2

The toolchain will be installed in the /usr/cross/powerpc-eabi directory.

3. Set a path to the /usr/cross/powerpc-eabi/bin: in the .bashrc file add the following:

```
export PATH=$PATH:/usr/cross/powerpc-eabi/bin
```

4. Test the toolchain installation:



5. Installing Eclipse IDE

Download the Eclipse IDE from here or get it form the CD:

http://download.ronetix.info/eclipse/eclipse.tar.bz2

The file eclipse.tar.bz2 includes:

- Eclipse SDK v3.2.1
- Embedded CDT v3.1 patched by Zylin
- Zylin plugin
- Java Virtual Machine v1.5.0_09



Set a path to the /usr/local/eclipse: in the .bashrc file add the following:

export PATH=\$PATH:/usr/local/eclipse

6. Installing an example

Download the MPC5554 example from here:

http://download.ronetix.info/examples/mpc5554_example.tar.gz_or get it from the CD.

Cd mkdir work cd work tar xvfz mpc5554_example.tar.gz

At this point you should be able to build, debug and run applications on embedded PowerPC targets. You can compile and debug the example manual, from the shell prompt or using Eclipse. The working with Eclipse is explained in "Section 5: Working with Eclipse" from this manual.

7. Compiling from the shell

Cd mpc5554demo/mpc_ram [linbox]\$ make

8. Debugging

To simplify the debugging there is a file '.gdbinit' in the project directory which is read when you run gdb/insight. Adjust the '~/work/mpc5554demo/mpc_ram/.gdbinit' to match the IP address of PEEDI.

From a X-console start the Insight debugger:

[linbox] powerpc-eabi-insight mpc_ram.elf

🗙 main.c - Source Window 🔹 🗖						
<u>F</u> ile <u>R</u> un <u>V</u> iew <u>C</u> ontrol <u>P</u> references <u>H</u> elp						
🍕 🕐 🕐 🕐 *0 😗 🕸 👗 \land 📾 🖧 📾 📲 🚳 🖿 👘						
main.c 💌 main 💌 SOURCE 💌						
128 int main(void) - 129 { maid (taphanatas)(and) (maid (t)(and)) ((maid and int) amounts)						
- 130 Vold (*pDnrystone)(Vold) = (Vold (*)(Vold)) ((Unsigned int) Af91F_Dr - 131 AT91F_DBGU_Printk("-I- Enter in main() section\n\r"); 132						
133 AT91F_DBGU_Printk("-I- no MMU, I+D Caches Disabled\n\r"); 134 AT91F_DisableICache();						
- 135 AT91F_DisableDCache(); - 136 AT91F_Dhrystone(); 137						
- 138 AT91F_DBGU_Printk("\n\r-I- no MMU, I Cache Enabled\n\r");						
Program stopped at line 130 20000d30 130						

The following descriptions discuss the use of the default debugger toolbar buttons.

The **Run** button, do not use this button. Instead of this use the "**Continue**" button

During the debugging process, the **Run** button turns into the **Stop** button to interrupt the debugging.





- The **Next** button steps to the next executable line of source code in the current file. Unlike the **Step** button, the **Next** button steps over called functions.
- The **Finish** button finishes execution of a current frame. If clicked while in a function, it finishes the function and returns to the line that called the function.

Download Performance

*0

0

0

By default, GDB versions previous than v6.5.50 (from 06.08.2006) download program code and data in small packets that are not necessarily a multiple of four bytes in length. This causes program download times to be slower than necessary, especially with PowerPC targets. There are two GDB internal variables that affect this. To improve GDB download performance, you should set the download write size to a binary value 16KB and the memory write packet size to a larger value to allow for packet overhead (+100 bytes is plenty). For example, to download 16KB at a time:

```
(gdb) set remote memory-write-packet-size fixed
(gdb) set remote memory-write-packet-size 16384
(gdb) set download-write-size 16128
```

Note:

No error is returned to GDB in case of an invalid 4 byte memory read access during



a period defined by **GDB_READ_INGNORE_TIME** in the target configuration file. This is because GDB refuses to connect to the target if it gets an error during this connection sequence. GDB tries to read the stack frame during the connection sequence and this may lead to invalid memory access.

4 Toolset installation on Windows

Please download the GNU toolchain installer from our web site, launch it and follow the instructions. <u>http://download.ronetix.info/toolchains/powerpc/ronetix-toolset-ppc.exe</u>

The setup will install:

By default, the Cygwin installer provided by Ronetix installs everything necessary for developing. This includes:

- Powerpc-eabi GNU tools (compiler, linker, and debugger)
- Eclipse IDE v3.2.1 and Zylin CDT plug-in
- Java run-time environment for the Eclipse
- ready-to-build examples
- a free TFTP server

5 Working with Eclipse

5.1 Adding a project

Eclipse (<u>http://www.eclipse.org</u>) is an open-source, Java based, powerful Integrated Development Environment (IDE). Adding the CDT plug-in (C/C++ Development Toolkit), you can edit and build C programs using the GNU compiler toolkit. A detailed C/C++ Development User Guide for Eclipse can be downloaded from <u>http://www.eclipse.org/cdt</u> or from the Ronetix site.

The Java run-time environment, Eclipse and the CDT plug-in are installed by the installer from Ronetix. Now start Eclipse IDE using the desktop icon, the "Workspace Launcher" dialog should appear, where you need to point the workspace folder. In our case point "C:\Program Files\Ronetix\examples\mpc5554\mpc5554demo", this way eclipse will automatically include the project files when we create a new project. Next click OK.

🗲 Workspace Launcher 🛛 🔀
Select a workspace
Eclipse Platform stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.
Workspace: C:\Program Files\Ronetix\examples\mpc5554\mpc5554demo 💌 📴
Use this as the default and do not ask again
OK Cancel

Now click at the "Workbench" icon at the right edge of the window.



The next window will look like:

듣 Java - Eclipse SDK				_ 🗆 🗙
<u>File Edit Source Refactor Navigate Search</u>	<u>Project R</u> un <u>W</u> indow <u>H</u> elp			
] 🗈 • 🗄 🛓 🏇 • 🔕 • 🧏 •] 🖉	瞪 🞯 •] 🕭 🔗] 🔂 •] 🧏 - 🍕 -	· *= + - =	·	🐉 Java
📕 Package Explorer 🗙 Hierarchy 🖵 🗖			🗖 🗖 📴 Outline	8 -0
			An outline is	not available.
	Problems 🛛 Javadoc Declaration			
	0 errors, 0 warnings, 0 infos		D 11	
		Resource	Path	Location
	L			
」□◆				🕋 🔮 🗷 🔮 🔶

Now you need to add as projects the mpc5554 example. Click File->New->Project. In the New Project dialog select Standard Make C Project and click Next:

E New Project	×
Select a wizard Create a new C Project which uses a simple makefile.	
<u>W</u> izards:	
type filter text	
Java Project Java Project from Existing Ant Buildfile Plug-in Project General C Managed Make C Project Standard Make C Project C C C C C C C C C C C C C C C C C C C	
⊘ < <u>B</u> ack <u>N</u> ext > Einish	Cancel

In the "New Project" dialog put "BasicMMU" for project name and click Finish. The BasicMMU already exists there so all of its files will be automatically imported in the new created Eclipse project.

E New Project	×
C++/Make Project Create a New C++ Project using 'make' to build it	G
Project name: mpc_ram	
Location: C:/Program Files/Ronetix/examples/mpc5554/mpc5554demo/mpc_ram	Browse
< Back Next > Finish	Cancel

Now Eclipse will ask you to change the perspective to C/C++, check remember my decision and click YES:

E Open Associated Perspective?	×
This kind of project is associated with the C/C++ perspective. Do y to open this perspective now?	you want
Remember my decision	. 1
<u>Y</u> es	

Now the Eclipse window should look like:

¢	C/C++ - Eclipse Platform										_ []	<
E	ile <u>E</u> dit Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject	<u>R</u> un <u>W</u> indow	Help									
1	📫 • 🖪 📥 🗟 - 🗎 🍪 • 🖓	• 🖻 • 🞯 •	\$*• • •	🌯 🕶 🛛 🙆	9 🔗 🖻	•] 🖢 -	图			😰 🔂 C/C+-	+	»
1	🖸 C/C++ Projects 🕱 🛛 Navigator 🛛 🗖)[- 0	1 20	utline 🛛 M	ake 🗖 🗖)
									An out	tline is not availa	ble.	1
Ľ	⊕ 🌮 mpc ram											
		🚼 Problems 🗙	Console P	roperties							\$ ~ □ □	
		0 errors, 1 warni	ng, 0 infos									
		Description A	(4.1)		Resource	e Path		Loc	ation			
		Error	gs (1 item) Launching exter	nal scanner i	fo dei MDC, ram			Unka	nown			
Ļ						1			_	1000		
	■ ♀											<u> </u>

Use the "Project" menu and remove the checkmark at "Build Automatically":



Open main.c by double clicking on it in the BasicMMU/src folder:



Now build the project:



At this point your project should be compiled and liked to a single executable "mpc_ram.elf" file. You can check its presence.

5.2 Configuring and working with the Eclipse built-in debugger

First click the arrow right to the "bug" button and next "Debug..."



This opens the Debug configuration dialog. Select Embedded debug (Native) and click the New button:



This will change the look of the window to:

E Debug						
Create, manage, and run con	figurations Name: mpc_ram Image: Main Main Source Project: Commands	ymmon				
<pre>#C Embedded debug (Cygwin) #C Embedded debug (Native) ##C Embedded debug (Native) ##C Embedded debug (Native)</pre>	mpc_ram C/C++ Application: mpc_ram.elf Search Project	Browse				
	Apply	Revert				
0	Debug	Close				

- If your window does not look like the previous, follow the next two steps:
- 1. Enter BasicMMU for name and click Project Browse to select the BasicMMU and click OK:

Froject Selection	
Choose a project to constr	ain the search for a program
😂 mpc_ram	
0	OK Cancel

2 .Next enter mpc_ram.elf for C/C++ Application.

🗲 Debug		×
Create, manage, and run con	figurations	-
Image: Second system Image: Second system Image: Secon	Name: mpc_ram Main Source Common Project: mpc_ram Browse C/C++ Application: Search Project Browse	
•	Appl <u>y</u> Re <u>v</u> ert	
0	Debug Close	

Now select the Debugger tab and Browse to locate "... \Ronetix\powerpc-eabi\bin\powerpc-eabi-gdb.exe":

🗲 Debug	×
Create, manage, and run con	figurations
Image: Second system Image: Second system Image: Secon	Name: mpc_ram Main Image: Commands Source Common Debugger: Embedded GDB Image: Stop on startup at: Advanced. Debugger Options Image: Stop on startup at: Advanced. GDB debugger: rpc-eabi-elf-gdb.exe Browse GDB command file: Browse (Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)
	Apply
0	Debug Close

Now change to the Commands tab and enter these commands considering your PEEDI IP address:

E Debug		
Create, manage, and run configurations		
Image: Second system Image: Second system <td>Name: mpc_ram Main Source Main Source Commands 'Initialize' commands target remote 192.168.3.56:2000 load si 'Run' commands</td>	Name: mpc_ram Main Source Main Source Commands 'Initialize' commands target remote 192.168.3.56:2000 load si 'Run' commands	
	Appl <u>y</u> Re <u>v</u> ert	
0	Debug Close	

These command will tell gdb:

- to connect to your PEEDI
- to load the project executable file to the target

Next clock Apply and Debug, this will start the debugger and Eclipse will ask you:

🦲 Conf	irm Perspective Switch	×	
?	This kind of launch is configured to open the Debug perspective when it suspends.		
	This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.		
	o you want to open this perspective now?		
⊠ Ren	nember my decision) <u>Y</u> es <u>N</u> o		

Check remember my decision and click YES.

Now the debugger is started:





Next click the Resume botton. The application is started and shortly it should break were the break point is:

E Debug - main c - Eclinse Platform			
File Edit Refactor Navigate Search Project Run Window Help			
] 📬 • 👷 💩 🚵 🏇 • O • O₄ • 🥭 🔗 Ga • ½ → S	$\bullet \leftarrow \diamondsuit \bullet \bullet \bullet$	🖹 🏇 Debug 🛛 👋	
🌮 Debug 🗙 📃 🗖	🕪= Variables 🛛 Breakpoints Re	gisters Modules	
🗸 🚸 🕩 II 🔳 M 🕄 🐼 La 🗮 🗷 🖬 🗸		🔄 📲 📄 💕 💥 🦓 🏹	
⊡…≢¢ mpc_ram [Embedded debug (Native)]	Name	Value	
🛱 💮 🚱 Embedded GDB (06-12-13 10:08) (Suspended)	(×)= cnt	1073811424	
E 🔐 Thread [0] (Suspended: Breakpoint hit.)			
I main() at C:\Program Files\Ronetix\examples\mpc5554\mpc5554			
C:\Program Files\Ronetix\powerpc-eabi-elf\bin\powerpc-eabi-elf-gdb.exe			
C:\Program Files\Ronetix\examples\mpc5554\mpc5554demo\mpc_ram\mp			
		<u> </u>	
i main.c ⊠start.s		Disassembly 🛛 🗖 🗖	
/* The code located here in "main()" may be replac	ed with */ 🔺 /*	your application code. 🔺	
/* your application code.	*/	int cnt = 0;	
int cnt = 0;	⇒0x4	40000014 <main+20>: li</main+20>	
init_LED2 (PIN_EMIOS12, PIN_EMIOS13); // Init	ialize the GPIC Ox4	40000018 <main+24>: stw 🔜</main+24>	
		init_LED2 (PIN_EMIOS12,	
/* interrupt configuration and use example	/* The following function call, when commented out, will bypass t 0x4000001c <main+28>: 1i</main+28>		
xcntn xmnl():	0x4	40000020 < main+322. 11	
/* The following function call uses ".data" and ".	sdata" elements 🗾 🛛 /*	The following function c	
Console 🕴 Tasks Memory	= ×	🔌 📴 🛃 🛃 🖬 - 🗂 🗖	
mpc_ram [Embedded debug (Native)] C:\Program Files\Ronetix\examples\mpc5554\mpc	:5554demo\mpc_ram\mpc_ram.elf (06-)	12-13 10:08)	
		<u> </u>	
		-	
T		F	
] 0*		🔗 💖 🔊 🌔 🔊	

At this point you can continue debugging, i.e. step, put breaks, resume, etc.

5.3 Using Insight debugger as an Eclipse external tool

I you prefer you can use Insight for a debugger when developing with Eclipse IDE.

For this reason you need to add it as an external tool by clicking Run->External Tools->External Tool...



In the "External Tools" dialog select Program and click the New button:

External Tools	×
Create, manage, an Run a program	d run configurations
type filter text	Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Filter' button to configure filtering options. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. Configure launch perspective settings from the <u>Perspectives</u> preference page.
0	Run Close

In the new dialog for name type "Insight", for location click Browse and navigate to "Ronetix\powerpceabi\bin\powerpc-eabi-insight.exe", for Working Directory click Browse Workspace and select BasicMMU and for Arguments type mpc_ram.elf.

🗲 External Tools	×
Create, manage, and run Run a program	configurations O
Image: Second system Lype filter text Image: Second system Image: Second system	Name: Inisight Image: Main Refresh Environment Common Location: Location: Image: State and S
0	<u>R</u> un Close

Now click on Common, and in the "Display in favorites menu" select Insight to add it to the Run menu and click Apply and Close:

External Tools	×	
Create, manage, and run configurations		
Run a program		
	Name: Insight	
type filter text	📄 Main 🚸 Refresh 🚾 Environment 💷 Common	
🖃 🖓 Ant Build	Save as	
New_configu		
	O Shared file:	
	Display in favorites menu Console Encoding	
	Content of the second sec	
	Standard Input and Output	
	Allocate Console (necessary for input)	
	File:	
	Browse Workspace Browse File System Variables	
	Append	
	✓ Launch in background	
	Apply Revert	
0	<u>R</u> un Close	

Now you have Insight in your Run menu.

ect	Run Window Help		
2 -	. 🗞 <u>R</u> un Last Launched	Ctrl+F11	1 🖉 🔗 1 🗟 • 1 🖢 •
	🍇 Debug Last Launched	F11	
\exists	Run His <u>t</u> ory	•	-
	Run A <u>s</u>	•	
	Ru <u>n</u>		
	Debug <u>H</u> istory	•	-
	Debug As	+	
	De <u>b</u> ug		_
	🍋 External Tools	•	🗞 <u>1</u> Insight
			Run As
			💁 External Tools
			Organize Fa <u>v</u> orites

You have just added the Insight.

You can follow the previous procedure to add any external tool you wish to use while developing. Now you can start Insight and debug the project.