



Microcross GNU X-Tools™

Professional Embedded Development Tools

Training Guide

August 2003 Edition

By

William A. Gatliff
Independent Consultant

And

James B. Calvin, Jr.
Microcross, Inc.

© 2003 Microcross, Inc.
Licensed Materials

MICROCROSS PRESS

Microcross GNU X-Tools™ Training Guide

by William A. Gatliff and James B. Calvin, Jr. (Authors)

August 2003 Edition

Copyright © 2003 by William A. Gatliff and Microcross, Inc.

All rights reserved. No part of this document may be reproduced, in any form or by any means, without prior written permission from the Authors.

The name GNU X-Tools™, Microcross logo, and Microcross Press are trademarks of Microcross, Inc.. Other product names and trademarks are the property of their respective holders.

Every precaution has been taken in the preparation of this document. The authors assume no responsibility for errors or omissions, or for damages resulting from the use of information contained therein.

Table of Contents

Preface	x
Prerequisites	x
Objectives.....	x
Procedures	x
Notation conventions	x
System requirements.....	xi
Acknowledgements.....	xi
1. GNU and the Law.....	1
"Free" vs. "Open Source"	1
Derivative Works and Anticipated Use.....	1
Not Legal Advice.....	2
The GNU General Public License, GPL	2
Derivative works of GPL articles must be released under the GPL.....	2
Users of GPL works are entitled to the source code for the work.....	2
Users are free to redistribute the work, or modified versions thereof.....	3
Users must know that they have these rights	3
The GNU Lesser General Public License, LGPL	3
Proprietary works	4
"A work that uses the library"	4
Users must know that they have these rights	5
The BSD License.....	5
Exercises.....	5
2. A Day In The Life of Embedded GNU.....	7
Create an application	7
Compile "hello, world!".....	7
Analyze "hello, world!"	8
Run "hello, world!" under simulation.....	10
Run "hello, world!" on target hardware.....	10
Translate "hello, world!" to other formats	12
3. Preparing to Build a GNU Toolchain	13
Build, Host and Target Machines	13
Prefix.....	14
Help	14
The configuration step	14
The build and installation steps.....	14
Microcross X-tools	15
The bootstrap GCC.....	15
4. Building an arm-elf GNU Toolchain	16
Acquire the source code.....	16
Select build, host and target specifications	16
Build binutils.....	17
Build a bootstrap gcc.....	18
Build newlib	18

Build gcc	19
Build gdb	19
Test the toolchain	20
Congratulations!.....	20
5. Native, Simulated, and Remote Debugging (gdb)	21
Native debugging.....	21
Debugging under simulation	22
Remote debugging.....	24
Other remote protocols.....	25
Remote debugging in a custom target	26
6. More Debugging Techniques (gdb)	27
Display and print	27
Set and set variable	27
x	28
Default radix	28
Registers	29
Breakpoints by file and line number	29
Breakpoints by address	30
Temporary breakpoints.....	30
Conditional breakpoints.....	30
Step and next with count.....	31
Stepi with count.....	31
Set step-mode	31
Until with location.....	31
Finish	32
Return	32
Backtrace	32
User-defined commands	32
Conditional commands	32
Command files	33
Apropos and help	33
Using tee with gdb.....	33
Tracepoints.....	34
Exercises	34
An initialization script for the Evaluator7T	34
The until and set variable commands.....	35
Stepping through assembly code	37
7. Graphical debugging (ddd)	40
Installing ddd.....	40
Using ddd for native debugging.....	40
Using ddd for remote debugging	41
8. Dissecting Applications (objdump)	42
Header information	42
Mixed listings	43
Disassembly	44
Symbols	46
Application size.....	46

Exercises.....	47
Analyze "hello, world!"	47
More analysis of "hello, world!"	48
9. Working with Libraries (ar).....	49
Using an archive	49
Linker archive searches.....	49
Creating an archive.....	50
Summarizing archive contents	50
Weak declarations.....	51
10. GNU Compiler Techniques (gcc)	52
Verbose output.....	52
Compilation and assembly	53
Debugging information	54
Warning levels	54
Preprocessor definitions	55
Dependency generation.....	56
Avoiding registers	56
Structure and enumeration packing.....	56
Assembler and Linker Options.....	57
Optimization	57
Automatic linking of startup code and libraries	57
Filename extensions	58
11. More GNU Compiler Techniques (gcc)	59
Object alignment.....	60
Packed objects	60
Memory section assignment.....	61
Symbol aliases	62
Declarations.....	63
Explicit register objects	64
Exercises.....	64
12. Linker Command Scripts.....	65
The MEMORY command.....	65
The SECTIONS command.....	66
Output section descriptions.....	66
Symbol assignment	68
Builtin functions.....	69
ABSOLUTE	69
ADDR.....	70
ALIGN	70
LOADADDR.....	71
SIZEOF.....	71
Referring to linker symbols from C	71
A simple example	72
A more comprehensive example	75
Assigning C objects to specific memory sections	73
Naming output sections.....	75
Handling initialized data.....	75

Generating a reset vector	76
A complete command script.....	77
Template scripts	78
Exercises	79
Construct a simple command script for the Evaluator7T	79
Locate data in the Evaluator7T's on-chip SRAM.....	81
Locate initialized data in internal SRAM	83
Locate a function in internal SRAM.....	83
The MEMORY command.....	83
13. Manipulating Application Images (objcopy)	84
Converting between file formats.....	84
Filling memory gaps.....	86
Extracting a memory section	88
Inserting a memory section.....	89
Relocating a memory section.....	90
Interleaving an application image	90
Exercises	92
Implement a reconfigurable string table	92
Interleave an application image.....	94
Merge two application images.....	94
14. Assembly Language Programming (gcc)	96
Inline assembly language.....	96
Operand constraints	100
Assembler templates	101
Output constraints.....	101
Input constraints	102
Clobber expressions	103
Multiple inputs and outputs	103
Optimization and inline assembly code	104
The <code>volatile</code> keyword	105
Type conversions in inline assembly code	105
The <code>inline</code> keyword.....	106
Introduction revisited	107
Programming entirely in assembly language	108
Invoking the assembler.....	108
Preprocessing	108
Assembler directives	109
Local symbol names	110
Declaring data	111
The special dot symbol	111
Machine dependencies	111
ARM Family	112
Hitachi H8 Family	112
Hitachi SH Family	112
Intel and Motorola.....	112
MIPS	112
Exercises	113

"Hello, world!" in inline assembly	113
"Hello, world!" in a standalone file	115
Invoking an external assembly function	115
The <code>inline</code> keyword	118
Duplicating common mistakes	118
15. Interrupt Handlers	119
ARM7TDMI	119
CPU32 interrupt stub	120
Assembly language modules	121
16. uC/OS-II	122
What is uCos?	122
17. Newlib	124
License	124
Printf() and iprintf() functions	124
Unix style API	124
Dynamic memory allocator	125
Libm	125
Reentrancy	125
Portability stubs	125
An embedded filesystem?!	125
18. Porting Newlib	126
Portability stubs	126
<code>_write_r</code> and <code>_read_r</code>	126
<code>_open_r</code>	127
<code>_close_r</code>	128
<code>_sbrk_r</code>	128
<code>errno</code>	129
Managing <code>_reent</code> structures	129
Reentrant <code>malloc()</code>	130
RTOS memory pools	131
<code>__env_lock</code> and <code>__env_unlock</code>	132
<code>_exit</code>	132
<code>getpid</code>	132
<code>_times_r</code>	133
<code>stdin</code> , <code>stdout</code> , and <code>stderr</code>	133
Exercises	134
19. The Macraigor Wiggler	135
Theory of operation	135
Operation	135
20. The Abatron BDI2000	137
Theory of operation	137
Operation	137
21. Debugging Agents	138
Debugging agents vs. JTAG et al	138
The GDB Remote Serial Protocol	139
"Host" vs. "Target"	139

Read Registers (g)	139
Write Registers (G)	140
Write Register n (P)	140
Read Memory (m).....	140
Write Memory (M)	140
Query Section Offsets (qOffsets)	140
Set Thread (H).....	140
Step (s).....	141
Continue (c)	141
Report Target Status (?)	141
Terminate Application (k)	142
Target Status Response (S)	142
Expedited Target Status Response (T)	142
Console Output (O)	142
Empty response	142
The <i>register file</i>	143
Breakpoint handler	143
Instruction stepping.....	144
Software-only instruction stepping	145
Stepping read-only code	147
Returning from exceptions	148
Debugging a debugging agent.....	149
Scripts and target initialization	149
The gdbstubs project.....	150
Exercises	150
Implement a debugging agent for the Evaluator7T.....	150
22. Startup Code and crt0.....	176
In the beginning	176
The <i>start</i> function	176
The <i>gccmain</i> function.....	177
Exercises	178
23. Gcc spec Files.....	179
What is a <i>spec</i> file?	179
To modify a <i>spec</i> file, or not?.....	179
The <i>spec</i> syntax.....	179
Creating and referring to strings	180
Renaming strings	181
Substitutions	181
Command line substitutions	181
Comparisons.....	182
Whitespace in <i>spec</i> files.....	182
Exercises	182
24. Using GNU for C++ Development.....	183
Global constructors and destructors.....	183
The <i>new</i> keyword.....	184
Compiling C++ programs.....	184

A. The GNU General Public License.....	185
B. The GNU Lesser General Public License.....	192
C. GNU Crossdevelopment Toolchain Build Script.....	201
D. Reconfiguring Newlib for Standalone ARM Operation.....	205
Problem description	205
Problem solution.....	205
E. BDI2000 Configuration Script for Evaluator 7T.....	207
About the author	210
Colophon	211