

Contents

1	Introduction	1
1.1	Demo Environment used for This Book	1
1.2	Book Contents	2
1.3	Conventions	2
1.4	Copyrights and Trademarks	3
2	Basic Concepts & Overview	5
2.1	Basic Terms	5
2.1.1	Target and Host	5
2.2	Real-time	5
2.3	Determinism – or Predictability?	6
2.3.1	Definition of Determinism	7
2.3.2	Definition of Predictability	7
2.3.3	Comparison	7
2.4	Portability	7
2.5	Development Strategies	8
2.5.1	Target based Development	8
2.5.2	Cross Development	8
2.6	Layered Architecture	9
2.6.1	Board Support Package — BSP	10
2.6.2	Kernel and Architecture-dependent Code	10
2.6.3	Application	11
2.7	Host Support Package HSP	11
2.7.1	Tool Sets and Tool Architecture	11
2.7.2	Target based Tools	17
2.8	Implementation of the Tool Set on the Different Hosts	18
2.8.1	UNIX Hosts - Solaris and HP/UX	18
2.8.2	Windows NT Hosts	18
2.8.3	All Hosts	18
2.9	Documentation	18
2.9.1	Tornado User’s Guide	19
2.9.2	VxWorks Programmer’s Guide	19
2.9.3	VxWorks Network — Programmer’s Guide	19

2.9.4	VxWorks Programmer's Reference	19
2.9.5	Tornado API Guide	19
2.9.6	Additional Documentation	20
2.9.7	GNU toolset documentation	20
2.9.8	Online Documentation	20
3	Setup	25
3.1	Installations	25
3.1.1	Before You Start - Think first!	25
3.1.2	Setting Up the Host	26
3.1.3	Installing the BSP	27
3.2	What is Where – The Directory Structure	31
3.2.1	<i>docs</i> directory	31
3.2.2	<i>host</i> directory	31
3.2.3	<i>setup</i> directory	31
3.2.4	<i>share</i> directory	32
3.2.5	<i>target</i> directory	32
3.2.6	But There is Still More - <i>.wind</i>	36
3.2.7	The Old Times — Before Tornado	36
3.3	Configuration Basics	36
3.3.1	Configuration Tool & Project Workspace	36
3.3.2	Rebuilding the Kernel	37
3.4	Different Goals, different Kernels	38
3.4.1	Bootrom	38
3.4.2	Kernel	38
4	Booting the Target	41
4.1	The bootrom	41
4.1.1	The Parameters	41
4.1.2	Bootrom Development – When and How	46
4.1.3	Network Boot	48
4.1.4	SCSI Boot	48
4.1.5	Shared Memory Network	57
4.2	Kernel Boot	59
4.2.1	Standard Kernel — <i>VxWorks</i>	59
4.2.2	Standalone Kernel — <i>vxWorks.st</i>	60
4.2.3	Rommable Kernels	60
4.3	Common Problems	60
5	Programming	63
5.1	Basic Concepts	63
5.1.1	Compatibility, Porting	63
5.2	Programming to Debug	64

5.2.1	Example Code	64
5.2.2	Compiling The Code	65
5.2.3	What happens when Downloading Code?	65
5.2.4	When to link code	65
5.3	Debugging	66
5.3.1	The <i>Crosswind</i> Debugger	66
5.3.2	Choosing Your Debug Environment	66
5.4	C++ and VxWorks	69
5.4.1	Wind Foundation Classes	69
5.4.2	tools.h++ and Booch Classes	70
5.5	Starting and Running your Application	70
5.5.1	Big Block - ONE package doing it All...	70
5.5.2	Startup Scripts	71
5.5.3	Dynamically Loading Applications and Starting Them	71
5.5.4	Usual Problems	72
5.6	Programming VME and PCI devices	72
5.6.1	Bus Devices	73
5.6.2	Programming Devices	77
5.7	VxWorks Additional Devices – Drivers	80
5.7.1	The Driver Concept of VxWorks	80
5.8	Interrupts in VxWorks	85
5.8.1	VMEbus Interrupts	86
5.8.2	Interrupt Internals	88
5.8.3	PCIBus Interrupts – What’s the Difference?	89
5.8.4	Emergency Interrupts – NMIs	90
5.9	Staying Portable	90
5.9.1	Reasoning	90
5.9.2	Considerations for staying Portable	91
5.9.3	POSIX.x	95
5.9.4	Conclusion	97
6	VxWorks Modules	99
6.1	How and Why does adding/removing Modules affect the Kernel Size?	99
6.1.1	Background – Symbol Tables	99
6.1.2	Linking the Modules	101
6.2	Basic Modules	101
6.2.1	Networking	101
6.2.2	SCSI	102
6.2.3	Target Agent and Target Shell	102
6.3	Optional Modules to add Functionality	102
6.3.1	FTP Server	102
6.3.2	NFS	102
6.3.3	NFS Server	103

6.3.4	Rlogin and Telnet	103
6.3.5	SENS	103
7	Important BSP Information	105
7.1	BSP Files	105
7.1.1	<i>Makefile</i>	105
7.1.2	<i>config.h</i>	107
7.1.3	<i><BSP_NAME>.h</i>	119
7.1.4	<i>sysSerial.c</i>	120
7.1.5	<i>sysScsi.c</i>	120
7.1.6	<i>sysVme.c</i>	120
7.1.7	<i>sysLib.c</i>	120
7.2	VxWorks System Files	122
7.2.1	<i>bootConfig.c</i>	122
7.2.2	<i>usrConfig.c</i>	122
7.2.3	<i>usrScsi.c, usrIde.c</i>	122
7.2.4	<i>usrNetwork.c</i>	122
7.2.5	<i>usrApplInit.c</i>	123
7.3	Gathering Additional Information	123
8	Tailoring Your Setup to Your Needs	125
8.1	Cleaning Up the Kernel Modules	125
8.2	Optimization	126
8.3	Profiling	128
8.3.1	RTLib – A Utility Library	129
8.3.2	Coverage Analysis Tools	129
8.4	WindView to Enhance System Performance	129
8.4.1	Instrumentation	129
8.4.2	Modes	130
8.4.3	Limitations	130
8.4.4	Things To Do After Using WindView	131
8.5	Tailoring the System to your Application	131
8.5.1	Application Structure	131
8.5.2	Hardware Setup	132
8.5.3	System Design	133
9	Troubleshooting VxWorks	135
9.1	Installation	135
9.2	Boot Process	135
9.2.1	Shared Memory Network	135
9.2.2	Download Issues	136
9.2.3	General Networking Issues	136
9.3	Running the Target	136

9.3.1	License Issues	136
9.3.2	Other Topics while Developing or Running the Target System	137
A	Important Defines and Modules	139
A.1	VxWorks Modules	139
A.1.1	Compatibility Packages	139
A.1.2	ANSI C support – default	140
A.1.3	BootP support – default	140
A.1.4	BSD socket support – default	140
A.1.5	Cache Support – default	141
A.1.6	C++ support – default disabled	141
A.1.7	Wind Foundation Classes – add-on	141
A.1.8	CodeTest Utility Library — add-on	142
A.1.9	Pre-Tornado Debug Facilities – default disabled	142
A.1.10	The DEMO application – default disabled	142
A.1.11	File systems – default disabled	142
A.1.12	Additional Network Interfaces – default disabled	143
A.1.13	UNIX-compatible Environment Variables – default	143
A.1.14	Basic Exception Handling – default	143
A.1.15	Floating Point and GCC Floating Point Libraries – default	143
A.1.16	Formatted I/O – default	144
A.1.17	FTP Server – default disabled	144
A.1.18	WindView – add-on	144
A.1.19	The I/O System – default	145
A.1.20	The Object Module Loader – default disabled	145
A.1.21	Logging Facilities – default	145
A.1.22	Full Featured Memory Manager – default	145
A.1.23	MIB 2 Support – default disabled	145
A.1.24	MMU support Levels - default and add-on	146
A.1.25	Message Queues – default	146
A.1.26	Networking – default	146
A.1.27	NFS Support Packages – default disabled	147
A.1.28	Pipes – default	148
A.1.29	POSIX Support - default disabled	148
A.1.30	Remote Serial Connections - PPP – default disabled	148
A.1.31	PROXY networks – default disabled	148
A.1.32	RAMdrv, A RAM disk driver – default disabled	149
A.1.33	Remote Debugging – default disabled	149
A.1.34	rlogin, allowing network access to your target – default disabled	149
A.1.35	Remote Procedure Calls, RPC – default disabled	150
A.1.36	Shell Security Package – default disabled	150
A.1.37	Socket Select() package – default	150
A.1.38	Different Semaphores – default	150

A.1.39	Target based shell – default disabled	151
A.1.40	Show Routines – default disabled	151
A.1.41	Signal facility – default	151
A.1.42	Serial Line IP Package – default disabled	151
A.1.43	Shared Memory Network, Communication via a backplane bus – default	151
A.1.44	Shared Memory Objects, VxMP – add-on	152
A.1.45	SNMP Agent – add-on	152
A.1.46	SPY, Task Monitoring Facility – default disabled	152
A.1.47	Standalone Symbol table – default disabled	152
A.1.48	Starting to do things right after booting, Startup Script – de- fault disabled	152
A.1.49	User-Readable Error states – default disabled	153
A.1.50	Standard I/O – default	153
A.1.51	STREAMS – add-on	153
A.1.52	Software Floating Point – default disabled as of Tornado 2 . . .	153
A.1.53	Symbol Table – default disabled	154
A.1.54	Task Hooks - default	154
A.1.55	Task Variables – default	154
A.1.56	TCP Debug Tools – default disabled	154
A.1.57	Telnet Style Remote Login to the target system – default disabled	154
A.1.58	TFTP Tools – default disabled	155
A.1.59	Function Execution Time Measurement Library – default . . .	155
A.1.60	Serial Devices – default	155
A.1.61	Object Module Unloader – default	155
A.1.62	Watchdogs – default	155
A.1.63	WDB, the Wind DeBug Agent – default	156
A.1.64	Zero Copying Sockets – default disabled	156
A.1.65	New features with Tornado 2 – default disabled	156
A.2	CPU Types	157
A.2.1	MC68000, MC68010, MC68020, MC68030, MC68040, MC68LC040, MC68060 – Motorola 680x0 Series	157
A.2.2	CPU32 – Motorola MC 68360 Series	157
A.2.3	SPARC, SPARClite – SUN Microsystems SPARC Architecture	158
A.2.4	I960CA, I960KA, I960KB, I960JX, I960HX – Intel i960 CPU Series	158
A.2.5	R3000, R4000, R4000, R4650, MIPS – MIPS Processors . . .	158
A.2.6	AM29030, AM29200, AM29035 – AM29xxx Series	158
A.2.7	PPC601, PPC602, PPC603, PPC604, PPC403, PPC505, PPC740, PPC750, PPC860, PPCEC603 – Motorola/IBM/Apple Pow- erPC Series	158
A.2.8	I80386, I80486, I80X86 – Intel x86 Series	158

A.2.9	SIMSPARCSUNOS, SIMSPARCSOLARIS, SIMHPPAHPUX, SIMNT – VxWorks Simulator Pseudo-Processor Series	158
B	Makefile Rules — A Short Introduction	159
B.1	Definitions And Rules	159
B.1.1	Additional Standard Targets	160
B.1.2	A Dependency Checker – <i>makedepend</i>	160
B.2	Example Codes	161
B.2.1	Example <i>Makefile</i>	161
B.2.2	<i>rtc.h</i>	162
B.2.3	<i>rtc.c</i>	162
B.3	Compiler Options	163
B.3.1	Compilation, Linking and Debugging	163
B.3.2	Optimization	164
C	Building you own Toolchain	165
C.1	Getting the Files	165
C.2	Rebuilding the Complete Toolchain	166
C.2.1	Building the Utility Toolchain	166
C.2.2	Rebuilding <i>gcc</i>	166
D	Pointers to Different Internet Sites	167
D.1	Specifications And References Online	167
D.2	Online VxWorks Pages And General Real-time Related Information	168
D.3	Programming Tools Online	168
D.4	Newsgroups	169
D.5	This Book	169

