BlueCat Linux Target Support Guide

DOC-0367-00

for Cogent CMA ARM Boards

Product names mentioned in *BlueCat Linux Target Support Guide for Cogent CMA ARM Boards* (variable) are trademarks of their respective manufacturers and are used here only for identification purposes.

Printed in the United States of America.

Contents

CHAPTER 1 *Overview*

The BlueCat Linux Target Support Guide (TSG) for the Cogent CMA ARM boards describes the BlueCat Linux boot procedure and the configuration of the prebuilt BlueCat Linux kernel contained in the arm_willow TSP distribution. It summarizes the demo systems supported in the arm_willow TSP. Chapters 4 and 5 list the supported demo systems and device drivers, respectively. Chapter 6 describes Thumb which is an extension of the ARM architecture. Chapter 7 provides an overview of the Automated Test System (ATS) and qualification of the ARM boards.

CHAPTER 2

Booting the Target

This chapter describes the BlueCat Linux boot procedure for the Cogent CMA ARM boards.

Booting BlueCat Linux on the ARM Board from ROM/Flash

Perform the following procedure to boot BlueCat Linux on the ARM board from Flash:

- Boot the OS Loader or the `install` demo system on the target from a network.
- Install your embedded BlueCat system into Flash using the OS Loader or the `install` demo system. Refer to the "Booting BlueCat Linux from Flash" section in the *BlueCat Linux User's Guide* for a detailed description of the installation procedure. It is important that the Linux kernel image resides at the beginning of Flash. Make sure you partition the Flash device accordingly using the `flash_fdisk` tool. For instance, the following command will create a single partition in which you can install a BlueCat Linux image of up to 1 MB:

  ```
  > exec flash_fdisk /dev/mtdchar0 0-10
  ```

- Reset the target board.
- Boot the executable `$BLUECAT_PREFIX/boot/arm_willow_fb` on the target from the network. As soon as this executable is booted on

the target, it jumps to Flash, thus passing control to the entry point of the BlueCat image installed in Flash. BlueCat boots from Flash.

NOTE: *The booting of `arm_willow_fb` could have been avoided if the SmartFirmware boot monitor had provided a command for jumping to Flash. Since such a command is not available, the `arm_willow_fb` executable is provided to demonstrate booting of BlueCat Linux on the ARM from Flash. On a custom ARM target, the functionality of `arm_willow_fb` can be mimicked by a firmware command that jumps to the offset 0x8000 in Flash.*

Booting BlueCat Linux on the ARM Board from a Network

From the SmartFirmware prompt, use the `boot` command to boot a BlueCat Linux embedded file system from a TFTP host. For instance:

```
ok boot net /tftpboot/ping.kdi
NVRAM mac_addr = 00:30:23:00:00:01
Command line: ramdisk_size=32768
...
```

NOTE: *In order for the boot command to work, the BOOTP daemon must be properly configured on the host for the embedded target. Please refer to the BlueCat Linux User's Guide for a description on how to set up the BOOTP daemon on the host.*

CHAPTER 3 *Default Kernel Configuration*

This chapter shows the configuration of the prebuilt BlueCat Linux kernel contained in the arm_willow TSP distribution.

Table 3-1: BlueCat Linux Default Configuration for the arm_willow TSP

Parameters	Table Number
System and Processor Type	Table 3-2
Code Maturity Level Options	Table 3-3
Loadable Module Support	Table 3-4
General Setup	Table 3-5
Plug-and-Play Support	Table 3-6
Block Devices	Table 3-7
Character Devices	Table 3-8
Video for Linux	Table 3-9
Joystick Support	Table 3-10
Ftape, the Floppy Tape Device Driver	Table 3-11
Networking Options	Table 3-12
Amateur Radio Support	Table 3-13
IrDA Subsystem	Table 3-14
Network Device Support	Table 3-15
ARCnet Devices	Table 3-16
Ethernet (10 or 100 Mbit)	Table 3-17
Token Ring Devices	Table 3-18
WAN Interfaces	Table 3-19
SCSI Support	Table 3-20
SCSI Low-Level Drivers	Table 3-21
Sound	Table 3-22

Table 3-1: BlueCat Linux Default Configuration for the arm_willow TSP (Continued)

Parameters	Table Number
Filesystems	Table 3-23
Network File Systems	Table 3-24
Partition Types	Table 3-25
Kernel Hacking	Table 3-26
LynuxWorks Messenger Support	Table 3-27
Modular Advanced Power Management	Table 3-28

Table 3-2: System and Processor Type

Option	Value	Description
CONFIG_ARCH_WILLOW	Y	CMA120(Willow)+ARM processor module
CONFIG_CPU_ARM7	Y	Optimize for ARM7 CPU
CONFIG_PAGESIZE_32	Not Set	Physical memory size

Table 3-3: Code Maturity Level Options

Option	Value	Description
CONFIG_EXPERIMENTAL	N	Prompt for development and/or incomplete code/drivers
CONFIG_ALIGNMENT_TRAP	N	Enable kernel-mode alignment trap handle (Experimental)
CONFIG_TEXT_SELECTIONS	N	Split text into discardable sections

Table 3-4: Loadable Module Support

Option	Value	Description
CONFIG_MODULES	Y	Enable loadable module support
CONFIG_MODVERSIONS	Y	Set version information on all symbols for modules
CONFIG_KMOD	Y	Kernel module loader

Table 3-5: General Setup

Option	Value	Description
CONFIG_NET	Y	Networking support
CONFIG_PCI	Y	PCI support
CONFIG_PCI_OLD_PROC	Y	Backward-compatible /proc/pci
CONFIG_BLUECAT_THUMB	N	BlueCat kernel support for THUMB binaries
CONFIG_BLUECAT_LOADER	N	BlueCat OS Loader
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat ignore printk
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat small memory footprint
CONFIG_BLUECAT_MEMSIZE	N	Memory sizing benchmark
CONFIG_SYSVIPC	N	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_SYSCTL	N	Systl support
CONFIG_NWFPE	Y	Math emulator
CONFIG_BINFMT_AOUT	N	Kernel support for a.out binaries
CONFIG_BINFMT_ELF	Y	Kernel support for ELF binaries
CONFIG_BINFMT_MISC	N	Kernel support for MISC binaries
CONFIG_ARTHUR	N	RISC OS personality
CONFIG_PARPORT	N	Parallel port support

Table 3-6: Plug-and-Play Support

Option	Value	Description
CONFIG_PNP	N	Plug and Play support

Table 3-7: Block Devices

Option	Value	Description
CONFIG_BLK_DEV_FD	N	Normal PC floppy disk support
CONFIG_BLK_DEV_IDE	N	Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
CONFIG_BLK_DEV_HD_ ONLY	N	Old hard disk (MFM/RLL/IDE) driver
CONFIG_BLK_DEV_LOOP	N	Loopback device support
CONFIG_BLK_DEV_NBD	N	Network block device support
CONFIG_BLK_DEV_MD	N	Multiple devices driver support
CONFIG_BLK_DEV_RAM	Y	RAM disk support
CONFIG_BLK_DEV_INITRD	N	Initial RAM disk (initrd) support
CONFIG_BLUECAT_RFS	Y	BlueCat RFS support
CONFIG_BLK_DEV_ GENERIC_FLASH_DOC	N	M-Systems DiskOnChip
CONFIG_BLK_DEV_XD	N	XT hard disk support
CONFIG_BLK_DEV_DAC960	N	Mylex DAC960/DAC1 100 PCI RAID Controller support
CONFIG_PARIDE_PARPORT	Y	Parallel port IDE device support
CONFIG_BLK_CPQ_DA	N	Compaq SMART2 support

Table 3-8: Character Devices

Option	Value	Description
CONFIG_VT	N	Virtual terminal
CONFIG_SERIAL	Y	Standard/generic (dumb) serial support
CONFIG_SERIAL_CONSOLE	Y	Support for console on serial port
CONFIG_SERIAL_ EXTENDED	N	Extended dumb serial driver options
CONFIG_SERIAL_ NONSTANDARD	N	Non-standard serial port support
CONFIG_UNIX98_PTYS	N	Unix98 PTY support
CONFIG_MOUSE	N	Mouse support (not serial mice)
CONFIG_QIC02_TAPE	N	QIC-02 tape support
CONFIG_WATCHDOG	N	Watchdog Timer Support

Table 3-8: Character Devices (Continued)

CONFIG_NVRAM	N	/dev/nvram support
CONFIG_RTC	N	Enhanced Real Time Clock Support
CONFIG_DTLK	N	Double Talk PC internal speech card support

Table 3-9: Video for Linux

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video for Linux

Table 3-10: Joystick Support

Option	Value	Description
CONFIG_JOYSTICK	N	Joystick support

Table 3-11: Ftape, the Floppy Tape Device Driver

Option	Value	Description
CONFIG_FTAPE	N	Ftape (QIC-80/Travan) support
CONFIG_FT_NORMAL_DEBUG	Not Set	Debugging output
CONFIG_FT_FULL_DEBUG	Not Set	Floppy tape controller

Table 3-12: Networking Options

Option	Value	Description
CONFIG_PACKET	Y	Packet socket
CONFIG_NETLINK	Y	Kernel/User netlink socket
CONFIG_RTNETLINK	Y	Routing messages
CONFIG_NETLINK_DEV	Y	Netlink device emulation
CONFIG_FIREWALL	N	Network firewalls
CONFIG_FILTER	Y	Socket Filtering
CONFIG_UNIX	Y	UNIX domain sockets
CONFIG_INET	Y	TCP/IP networking

Table 3-12: Networking Options (Continued)

CONFIG_IP_MULTICAST	Y	IP: multicasting
CONFIG_IP_ADVANCED_ ROUTER	N	IP: advanced router
CONFIG_IP_PNP	N	IP: kernel level autoconfiguration
CONFIG_IP_ROUTER	N	IP: optimize as router not host
CONFIG_NET_IPIP	N	IP: tunneling
CONFIG_NET_IPGRE	N	IP: GRE tunnels over IP
CONFIG_IP_MROUTE	N	IP: multicast routing
CONFIG_IP_ALIAS	N	IP: aliasing support
CONFIG_SYN_COOKIES	N	IP: TCP syncookie support (not enabled per default)
CONFIG_INET_RARP	N	IP: Reverse ARP
CONFIG_SKB_LARGE	N	IP: Allow large windows (not recommended if <16 MB of memory)
CONFIG_IPX	N	The IPX protocol
CONFIG_ATALK	N	AppleTalk DDP

Table 3-13: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur Radio support

Table 3-14: IrDA Subsystem

Option	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

Table 3-15: Network Device Support

Option	Value	Description
CONFIG_NETDEVICES	Y	Network device support
CONFIG_DUMMY	N	Dummy net driver support
CONFIG_EQUALIZER	N	EQL (serial line load balancing) support

Table 3-15: Network Device Support

CONFIG_NET_SB1000	N	General Instruments Surfboard 1000
CONFIG_FDDI	N	FDDI driver support
CONFIG_PPP	N	PPP (point-to-point) support
CONGIG_SLIP	N	SLIP (serial line) support
CONFIG_NET_RADIO	N	Wireless LAN (non-ham radio)
CONFIG_NET_FC	N	Fibre Channel driver support
CONFIG_SBNI	N	SBNI 12-xx support

Table 3-16: ARCnet Devices

Option	Value	Description
CONFIG_ARCNET	N	ARCnet support

Table 3-17: Ethernet (10 or 100 Mbit)

Option	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100 Mbit)
CONFIG_ARM_AM79C961A	N	AM79C961A support
CONFIG_NET_VENDOR_ 3COM	N	3COM cards
CONFIG_LANCE	N	AMD LANCE and PCnet (AT 1500 and NE2100) support
CONFIG_NET_VENDOR_ SMC	N	Western Digital/SMC cards
CONFIG_NET_VENDOR_ RACAL	N	Racal-Interlan (micom) NI cards
CONFIG_NET_ISA	N	Other ISA cards
CONFIG_SK_G16	N	SK-G16 support
CONFIG_NET_EISA	Y	EISA, VLB, PCI and on board controllers
CONFIG_PCNET32	N	AMD PCnet32 (VLB and PCI) support
CONFIG_APRICOT	N	Apricot Xen-II on board Ethernet
CONFIG_CS89x0	N	CS89x0 support
CONFIG_DM9102	N	DM9102 PCI Fast Ethernet Adapter support (exp)

Table 3-17: Ethernet (10 or 100 Mbit) (Continued)

CONFIG_DE4X5	N	Generic DECchip & DIGITAL EtherWORKS PCI/EISA
CONFIG_DEC_ELCP	N	DECchip Tulip (dc21x4x) PCI support
CONFIG_DGRS	N	Digi Intl. RightSwitch SE-Xsupport
CONFIG_EEXPRESS_PRO100	Y	Ether ExpressPro/100 support
CONFIG_NE2K_PCI	N	PCI NE2000 support
CONFIG_TLAN	N	TI ThunderLAN support
CONFIG_VIA_RHINE	N	VIA Rhine support
CONFIG_NET_POCKET	N	Pocket and portable adaptors

Table 3-18: Token Ring Devices

Option	Value	Description
CONFIG_TR	N	Token Ring driver support

Table 3-19: WAN Interfaces

Option	Value	Description
CONFIG_HOSTESS_SV11	N	Comtrol Hostess SV-11 support
CONFIG_COSA	N	COSA/SRP sync serial boards support
CONFIG_SEALEVEL_4021	N	Sealevel Systems 4021 support
CONFIG_DLCI	N	Frame relay DLCI support
CONFIG_WAN_DRIVERS	N	WAN drivers

Table 3-20: SCSI Support

Option	Value	Description
CONFIG_SCSI	Y	SCSI support
CONFIG_BLK_DEV_SD	Y	SCSI disk support
CONFIG_BLK_DEV_ST	N	SCSI tape support
CONFIG_BLK_DEV_SR	N	SCSI CD-ROM support
CONFIG_CHR_DEV_SG	N	SCSI generic support

Table 3-20: SCSI Support (Continued)

CONFIG_SCSI_MULTI_LUN	N	Probe all LUNs on each SCSI device
CONFIG_SCSI_CONSTANTS	N	Verbose SCSI error reporting (kernel size +=12K)
CONFIG_SCSI_LOGGING	N	SCSI logging facility

Table 3-21: SCSI Low-Level Drivers

Option	Value	Description
CONFIG_SCSI_7000FASST	N	7000FAST SCSI support
CONFIG_SCSI_ACARD	N	ACARD SCSI support
CONFIG_SCSI_AHA152X	N	Adaptec AHA152X/2825 support
CONFIG_SCSI_AHA1542	N	Adaptec AHA1542 support
CONFIG_SCSI_AHA1740	N	Adaptec AHA1740 support
CONFIG_SCSI_AIC7XXX	N	Adaptec AIC7xxx support
CONFIG_SCSI_IPS	N	IBM ServeRAID support
CONFIG_SCSI_ADVANSYS	N	AdvanSys SCSI support
CONFIG_SCSI_IN2000	N	Always IN2000 SCSI support
CONFIG_SCSI_AM53C974	N	AM53/79C974 PCI SCSI support
CONFIG_SCSI_MEGARAID	N	AMI MegaRAID support
CONFIG_SCSI_BUSLOGIC	N	BusLogic SCSI support
CONFIG_SCSI_DTC3280	N	DTC3180/3280 SCSI support
CONFIG_SCSI_EATA	N	EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support
CONFIG_SCSI_EATA_DMA	N	EATA-DMA [Obsolete] (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphatronix) support
CONFIG_SCSI_EATA_PIO	N	EATA-PIO (old DPT PM2001, PM2023A) support
CONFIG_SCSI_FUTURE_DOMAIN	N	Future Domain 16xx SCSI/AHA-2920 support
CONFIG_SCSI_GDTH	N	GDT SCSI Disk Array Controller support
CONFIG_SCSI_GENERIC_NCR5380	N	Generic NCR5380/53c400 SCSI support

Table 3-21: SCSI Low-Level Drivers (Continued)

CONFIG_SCSI_G_NCR5380_PORT	N	NCR5380/53c400 mapping method (use Port for T130B) (PORT)
CONFIG_SCSI_G_NCR5380_MEM	N	NCR5380/53c400 mapping method (use Port for T130B) (MEM)
CONFIG_SCSI_INITIO	N	Initio 9100U(W) support
CONFIG_SCSI_INIA100	N	Initio INI-A100U2W support
CONFIG_SCSI_NCR53C406A	N	NCR53c406a SCSI support
CONFIG_SCSI_SYM53C416	N	Symbios 53c416 SCSI support
CONFIG_SCSI_NCR53C7xx	N	NCR53c7, 8xx SCSI support
CONFIG_SCSI_NCR53C8xx	N	NCR53C8XX SCSI support
CONFIG_SCSI_SYM53C8xx	Y	SYM53C8XX SCSI support
CONFIG_SCSI_NCR53C8xx_DEFAULT_TAGS	8	Default tagged command queue depth
CONFIG_SCSI_NCR53C8xx_MAX_TAGS	32	Maximum number of queued commands
CONFIG_SCSI_NCR53C8xx_SYNC	20	Synchronous transfers frequency in MHz
CONFIG_SCSI_NCR53C8xx_PROFILE	N	Enable profiling
CONFIG_SCSI_NCR53C8xx_IOMAPPED	Y	Use normal IO
CONFIG_SCSI_NCR53C8xx_PQS_PDS	N	Include support for the NCR PQS/PDS SCSI card
CONFIG_SCSI_NCR53C8xx_SYNBIOS_COMPAT	N	Assume boards are SYMBIOS compatible
CONFIG_SCSI_PAS16	N	PAS16 SCSI support
CONFIG_SCSI_PCI2000	N	PCI2000 support
CONFIG_SCSI_PCI2220I	N	PCI2220i support
CONFIG_SCSI_PSI240I	N	PSI240i support
CONFIG_SCSI_QLOGIC_FAS	N	Qlogic FAS SCSI support
CONFIG_SCSI_QLOGIC_ISP	N	Qlogic ISP SCSI support
CONFIG_SCSI_QLOGIC_FC	N	Qlogic ISP FC SCSI support
CONFIG_SCSI_SEAGATE	N	Seagate ST-02 and Future Domain TMC-8xx SCSI support
CONFIG_SCSI_DC390T	N	Tekram DC390(T) and Am53/79C974 SCSI support

Table 3-21: SCSI Low-Level Drivers (Continued)

CONFIG_SCSI_T128	N	Trantor T128/T128F/T228 SCSI support
CONFIG_SCSI_U14_34F	N	UltraStor 14F/34F support
CONFIG_SCSI_ULTRASTOR	N	UltraStor SCSI support

Table 3-22: Sound

Option	Value	Description
CONFIG_SOUND	N	Sound support

Table 3-23: Filesystems

Option	Value	Description
CONFIG_QUOTA	N	Quota support
CONFIG_AUTOFS_FS	N	Kernel automounter support
CONFIG_AFFS_FS	N	Amiga FFS filesystem support
CONFIG_HFS_FS	N	Apple Macintosh filesystem support (exp)
CONFIG_FAT_FS	N	DOS FAT fs support
CONFIG_ISO9660_FS	N	ISO 9660 CDROM filesystem support
CONFIG_MINIX_FS	Y	Minix fs support
CONFIG_NTFS_FS	N	NTFS filesystem support (read only)
CONFIG_HPFS_FS	N	OS/2 HPFS filesystem support (read only
CONFIG_PROC_FS	Y	/proc filesystem support
CONFIG_ROMFS_FS	N	ROM filesystem support
CONFIG_EXT2_FS	Y	Second extended fs support
CONFIG_SYSV_FS	N	System V and Coherent filesystem support
CONFIG_UFS_FS	N	UFS filesystem support
CONFIG_BLUECAT_FFS	N	BlueCat Linux Flash File System support

Table 3-24: Network File Systems

Option	Value	Description
CONFIG_CODA_FS	N	Coda filesystem support (advanced network fs)
CONFIG_NFS_FS	N	NFS filesystem support
CONFIG_SMB_FS	N	SMB filesystem support (to mount WfW shares etc.)
CONFIG_NCP_FS	N	NCP filesystem support (to mount NetWare volumes)

Table 3-25: Partition Types

Option	Value	Description
CONFIG_BSD_DISKLABEL	N	BSD disklabel (BSD partition tables) support
CONFIG_MAC_PARTITION	N	Macintosh partition map support
CONFIG_SMD_DISKLABEL	N	SMD disklabel (Sun partition tables) support
CONFIG_SOLARIS_X86_PARTITION	N	Solaris (x86) partition table support

Table 3-26: Kernel Hacking

Option	Value	Description
CONFIG_FRAME_POINTER	N	Compile kernel with frame pointer (for useful debugging)
CONFIG_DEBUG_ERRORS	N	Verbose kernel error messages
CONFIG_DEBUG_USER	Y	Verbose user fault messages
CONFIG_DEBUG_INFO	N	Include debugging information in kernel binary
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_BLUECAT_KDBG	N	Include kdbg kernel debugger
CONFIG_BLUECAT_KDBG_TTYS0	Not set	Serial console device

Table 3-27: LynuxWorks Messenger Support

Option	Value	Description
CONFIG_BLUECAT_IOPMAN	N	Enable Lynx IOP Manager support
CONFIG_BLUECAT_MSNG	N	Enable Messenger Support

Table 3-28: Modular Advanced Power Management

Option	Value	Description
CONFIG_BLUECAT_APM	N	Modular Advanced Power Management support

CHAPTER 4 *Supported Demo Systems*

The following table shows the demo systems supported by the arm_willow TSP. Boot devices supported by the prebuilt demo systems included in your distribution are shown.

Table 4-1: Demo Systems Supported by the arm_willow TSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
`caffeine`	Network using the OS Loader, Network using the firmware	2684 KB	16000 KB
`default`	Network using the OS Loader, Network using the firmware	1497 KB	7500 KB
`disk`	Network using the OS Loader, Network using the firmware	1513 KB	8000 KB
`ffs`	Network using the OS Loader, Network using the firmware	1252 KB	6500 KB
`ftp`	Network using the OS Loader, Network using the firmware	1736 KB	9500 KB
`gdb`	Network using the OS Loader, Network using the firmware	1271 KB	7500 KB
`gdb_thumb`	Network using the OS Loader, Network using the firmware	1496 KB	8000 KB

Table 4-1: Demo Systems Supported by the arm_willow TSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
gnutar	Network using the OS Loader, Network using the firmware	1354 KB	7000 KB
hello	Network using the OS Loader, Network using the firmware	578 KB	3500 KB
hello_dynamic_thumb	Network using the OS Loader, Network using the firmware	1091 KB	6500 KB
hello_thumb	Network using the OS Loader, Network using the firmware	584 KB	4000 KB
install	Network using the OS Loader, Network using the firmware	1612 KB	8500 KB
kdbg	Network using the OS Loader, Network using the firmware	1266 KB	6500 KB
loadkeys	Network using the OS Loader, Network using the firmware	1573 KB	8000 KB
mapm	Network using the OS Loader, Network using the firmware	1182 KB	6500 KB
memsize	Network using the OS Loader, Network using the firmware	1262 KB	7000 KB
modular	Network using the OS Loader, Network using the firmware	1258 KB	7000 KB
msng_exmpl	Network using the OS Loader, Network using the firmware	1228 KB	7500 KB

Table 4-1: Demo Systems Supported by the arm_willow TSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
`multi_user`	Network using the OS Loader, Network using the firmware	2094 KB	10000 KB
`multi_user_net`	Network using the OS Loader, Network using the firmware	2511 KB	12000 KB
`nfsroot`	Network using the OS Loader, Network using the firmware	580 KB	4500 KB
`osloader`	Network using firmware	809 KB	19000 KB
`ping`	Network using the OS Loader, Network using the firmware	1338 KB	7500 KB
`rcp`	Network using the OS Loader, Network using the firmware	1479 KB	8500 KB
`rlogin`	Network using the OS Loader, Network using the firmware	1619 KB	9500 KB
`rootfs`	Network using the OS Loader, Network using the firmware	520 KB	3500 KB
`shell`	Network using the OS Loader, Network using the firmware	1301 KB	6900 KB
`shell_thumb`	Network using the OS Loader, Network using the firmware	1934 KB	9000 KB
`tcl`	Network using the OS Loader, Network using the firmware	1430 KB	7500 KB

Table 4-1: Demo Systems Supported by the arm_willow TSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
`tcpdump`	Network using the OS Loader, Network using the firmware	1436 KB	8000 KB
`tutorial`	Network using the OS Loader, Network using the firmware	1323 KB	7500 KB
`xclock`	Network using the OS Loader, Network using the firmware	3204 KB	16500 KB
`xdemo1`	Network using the OS Loader, Network using the firmware	3167 KB	16500 KB
`xdemo2`	Network using the OS Loader, Network using the firmware	3716 KB	19500 KB

CHAPTER 5 *Supported Device Drivers*

The following table shows the device drivers supported by the arm_willow TSP.

Table 5-1: The Device Drivers Supported by the arm_willow TSP

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Ethernet Controller Intel GD82559/ER	`eepro100.c`	`drivers/net/*.c`	`CONFIG_EEXPRESS_PRO100`	
SCSI Controller Symbios Logic 53C810	`sym53c8xx.c`	`drivers/scsi/sym53c8xx.c`	CONFIG_SCSI CONFIG_BLK_DEV_SD CONFIG_SCSI_SVM53C8XX CONFIG_SCSI_NCR53C8XX_DEFAULT_TAGS=8 CONFIG_SCSI_NCR53C8XX_MAX_TAGS=32 CONFIG_SCSI_NCR53C8XX_SYNC=20 CONFIG_SCSI_NCR53C8XX_IOMAPPED	
Keyboard/ Mouse PS/2	`pc_keyb.c`	`arch/arm/special/pc_keyb.c`	CONFIG_MOUSE CONFIG_PSMOUSE	
Serial Ports ST16c552, ST16c2552, (16550A compatible), UARTs.	Standard serial driver	`arch/arm/special/serial.c`	CONFIG_SERIAL	

Table 5-1: The Device Drivers Supported by the arm_willow

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Video B69000 LCD/CRT	Standard PC video console driver	`drivers/video/ chipsfb_69000.c`	CONFIG_FB CONFIG_DUMMY_ CONSOLE CONFIG_FB_CT69000 CONFIG_FBCON_ ADVANCED CONFIG_FBCON_CFB8	Graphic modes supported by X Windows.
Parallel Ports	Parallel port driver	`drivers/misc/ parport_ willow.c`	CONFIG_PARPORT CONFIG_PARPORT_ WILLOW	Driver not tested
CardBus Controller PD6832	PCMCIA Card Service	`pcmcia-cs-3.0.14/`		Driver not tested

CHAPTER 6 *Thumb Support Software*

Thumb is an extension of the ARM architecture. It has 36 instruction formats, which are drawn from the standard 32-bit ARM instruction set. These have been recoded into 16-bit wide op-codes. This results in very high code density, as Thumb instructions are half the width of ARM instructions. On execution, these new 16-bit Thumb op-codes are decompressed by the processor to their ARM instruction set equivalents, which are then run on an ARM core.

Thumb is not just another mixed instruction set concept. Thumb aware cores have two separate instruction sets. This is a unique advantage, as it provides the designer with all the power of the ARM's 32-bit instruction set, while maintaining the code-size advantages of the Thumb instruction set. The fact that the two instruction sets are distinct results in extremely simple decoding logic. This in turn enables small silicon size, and maintains the ARM's industry leading low-power capability and MIPS/Watt performance.

Thumb-aware cores, such as the ARM7TDMI, have the full 32-bit architecture, which is similar to the ARM. Hence, the designer retains 32-bit RISC performance. The combination of the two instruction sets, running on a 32-bit Thumb-aware core, provides an effective solution for the code-size and performance issues of 16-bit systems.

NOTE: *Not all ARM cores contain support for Thumb. For example, the StrongARM SA-110, supported by BlueCat Linux, does not contain the Thumb extension.*

Installing BlueCat Linux Thumb Support Software

The BlueCat Linux Thumb support software is distributed on a separate CD-ROM. The Thumb distribution CD-ROM is used to install Thumb support on top of the BlueCat Linux development environment that is installed on the host. Refer to the BlueCat Linux User's Manual for a detailed description of the BlueCat Linux installation procedure. Step through the following procedure to install the Thumb support software:

- Insert the Thumb distribution CD-ROM in the CD-ROM drive on your development host.
- If your host does not auto-mount the CD-ROM, mount the CD-ROM on the host.
- Go to the top of your BlueCat Linux installation for the ARM target. Enable the BlueCat execution environment. For instance:

  ```
  $ cd $HOME/bluecat
  $ . SETUP.sh
  BlueCat:$
  ```

- Run the `install` installation script residing at the top of the Thumb distribution CD-ROM to install Thumb support in your BlueCat Linux development environment.

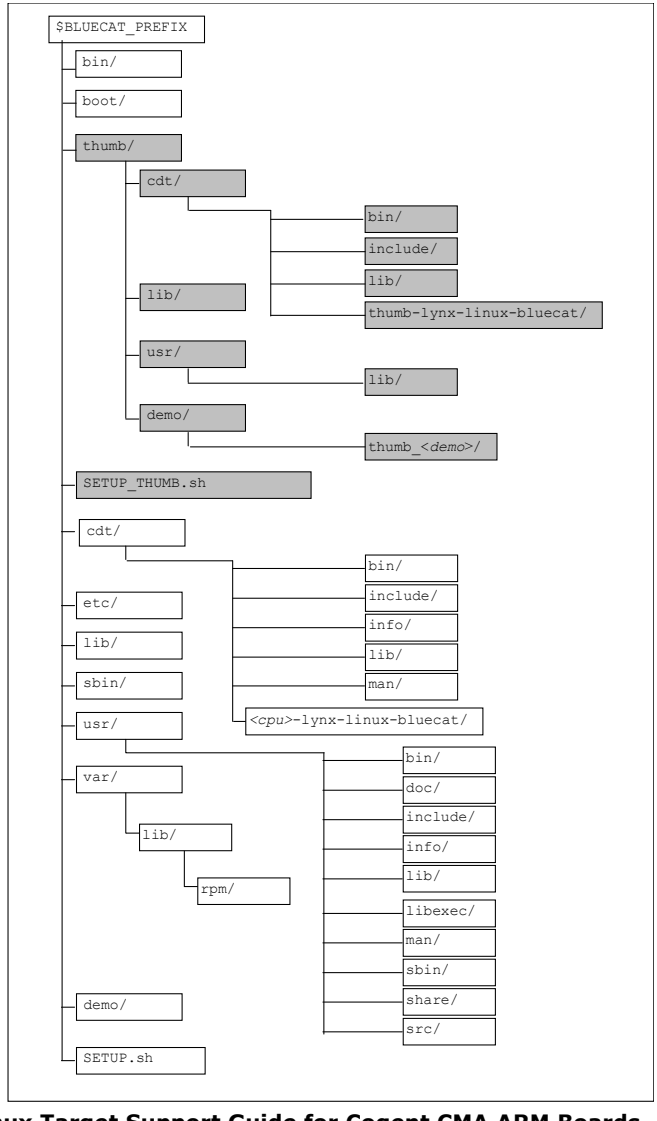
  For instance, assuming the CD-ROM is mounted at `/mnt/cdrom`, do the following:

  ```
  BlueCat:$ /mnt/cdrom/install
  ```

Upon successful completion of the installation, the Thumb support is installed in your BlueCat Linux development environment on the host.

Thumb Support Directory Structure Overview

The installation procedure described above results in the creation of the following Thumb-specific files and directories, shown in gray in the following figure.



The following table briefly describes the key components of the Thumb
specific directories in the BlueCat Linux tree:

Node	Description
$BLUECAT_PREFIX	installation directory
- thumb/	Thumb specific subtree
-- cdt/	Thumb cross-development tools
--- bin/	Thumb cross-development binaries
--- include/	Thumb cross-development `include` files
--- lib/	Thumb cross-development libraries
--- thumb-lynx-lynix-bluecat/	Thumb cross-development binaries
-- lib/	Thumb libraries
-- usr/	Thumb top-level `usr` directory
--- lib/	more Thumb libraries
-- demo/	BlueCat Linux configurations
--- thumb-<demo>/	Thumb specific demo configurations
SETUP_THUMB.sh	shell script for setting up the Thumb environment

The `thumb/cdt` subtree contains the Thumb cross-development tools. The `thumb/lib` subtree contains the Thumb target libraries. The `thumb/demo` subtree contains the Thumb specific demo systems.

Thumb Cross-Development Environment

The Thumb cross-development environment does not replace the cross-development environment for the ARM mode. Instead, the cross-development tools for both execution modes are available in parallel.

The Thumb cross-development tools are available in `$BLUECAT_PREFIX/thumb/cdt` in your BlueCat Linux development tree.

The GNU toolkit does not support switching between compiling for the ARM instruction set and the Thumb instruction set on anything other than a per-file basis. There are two completely different compilers. One produces the ARM assembler instructions and the other produces the Thumb assembler instructions.

The standard development tools for the ARM mode are default, while the Thumb development tools are optional. This means that the `gcc (cc, egcs)` command starts the standard ARM compiler. To run the Thumb compiler you can choose one of the following options:

- Enter a command specifying the full path to the compiler:

  ```
  $BLUECAT_PREFIX/thumb/cdt/thumb-lynx-linux-
  bluecat/bin/gcc
  ```

- Use one of the Thumb-specific commands `tcc`, `tgcc`, or `tegcs` to start the Thumb compiler. These Thumb-specific tools reside in the BlueCat tree along with the standard cross development tools in the `$BLUECAT_PREFIX/cdt/arm-lynx-linux-bluecat/bin` directory. The commands are symbolic links to the actual Thumb tools. For instance:

  ```
  tcc -> ../../../thumb/cdt/thumb-lynx-linux-
  bluecat/bin/cc
  tegcs -> ../../../thumb/cdt/thumb-lynx-linux-
  bluecat/bin/egcs
  tgcc -> ../../../thumb/cdt/thumb-lynx-linux-
  bluecat/bin/gcc
  ```

  The above Thumb-specific commands are available to the user immediately after the BlueCat Thumb specific `SETUP_THUMB.sh` script is executed.

  Run the `SETUP_THUMB.sh` script from the BlueCat root directory:

  ```
  $ cd $BLUECAT_PREFIX
  $ . SETUP_THUMB.sh
  BlueCat:$
  ```

Once the above script is executed, the standard commands `cc`, `gcc`, or `egcs` run the Thumb tools.

A new option, `--thumb-bc,` has been added to the standard linker `ld`. This option is used to mark the executable as a Thumb application. In order to produce a Thumb executable from the object files, you must specify the —`thumb-bc` option in the command line. Use the –`static` option to build a statically linked Thumb application. For instance:

```
ld ... -o thumbexec ... file1.o file2.o --thumb-bc -static
```

If the Thumb executable is linked by calling the Thumb specific `tgcc`, the `--thumb-bc` option is generated and passed to the `ld` command automatically.

A new option –`mthumb-bc` has been added to the standard assembler `as` to specify the Thumb assembler instructions in the input file.

Placing Thumb Applications onto the Target

You can combine ARM and Thumb applications on the target and execute them in parallel. Use the `mkrootfs` cross development tools to put Thumb applications in a file system installed on the target.

Thumb Support in the Kernel

You must enable Thumb support in the BlueCat Linux kernel to run Thumb applications on the target. The Thumb support is configured in the kernel using the kernel configuration option `CONFIG_BLUECAT_THUMB`.

Use the standard kernel configuration tools (for instance, `make xconfig`) to enable the Thumb support in the kernel. From the main menu go to `General Setup` and enable the `BlueCat kernel support for the Thumb binaries` option.

Thumb Demo Systems

The demo systems residing in `$BLUECAT_PREFIX/thumb/demo` describe the build and execution of Thumb applications on the 7T ARM target.

- The `hello_thumb` demo system shows the execution of an init-like Thumb application on the target.
- The `shell_thumb` demo system describes the execution of a Thumb application in parallel with ARM applications. The demo system starts the standard ARM-mode shell. `hello_world` is a Thumb demo that verifies the execution of Thumb applications in parallel with ARM applications.
- The `hello_dynamic_thumb` demo shows the dynamically- linked `hello_world` application.
- The `gdb_thumb` demo system shows the debugging of a simple Thumb application with the Thumb GDB debugger.

CHAPTER 7

Target Testing and Qualification

This chapter shows the results of the ATS testing and qualification of BlueCat Linux for the Cogent CMA ARM boards. For a detailed description of the BlueCat Linux Test Suite and Automated Test System (ATS) refer to the *BlueCat Linux User's Guide*.

ATS Suites Test Results

All supported ATS Suites were run on Cogent CMA ARM boards. The test results are shown in the chapter titled "BlueCat Linux Testing and Qualification" in the *BlueCat Linux User's Guide*.

Real-Time Performance

This section summarizes the results of the BlueCat Linux real-time performance test suite, which simulates a real-world system environment and measures the Interrupt Response and Task Response times.

The following three tables summarize the actual results of real-time performance measurements for Cogent CMA ARM boards. The results were obtained on the following hardware system:

- CPU - EP7211 (ARM 720T Core) at 33 MHz
- L1 Cache - 8 KB
- L2 Cache - None
- RAM – 32 MB
- Disk – SCSI 53c810A
- Ethernet – Intel i82559ER

Table 7-1: Real-Time Performance - Configuration 1

Configuration schedule policy: fifo schedule priority: 99 background load: network (ping -f)	
Interrupt Response: Best Measured: Average: Worst Measured:	 38 us 104 us 1,342 us
Task Response: Best Measured: Average: Worst Measured:	 415 us 1,331 us 168,276 us

Table 7-2: Real-Time Performance - Configuration 2

Configuration schedule policy: fifo schedule priority: 99 background load: disk (mkfs)	
Interrupt Response: Best Measured: Average: Worst Measured:	 40 us 77 us 2,177 us
Task Response: Best Measured: Average: Worst Measured:	 414 us 1,164 us 139,086 us

Table 7-3: Real-Time Performance - Configuration 3

Configuration schedule policy: fifo schedule priority: 99 background load: none	
Interrupt Response: Best Measured: Average: Worst Measured:	 37 us 47 us 165 us
Task Response: Best Measured: Average: Worst Measured:	 382 us 527 us 27,743 us