By David Marsh, Contributing Technical Editor

LINUX SEEMS TO HAVE BEEN THE "NEXT BIG THING" IN EMBEDDED SYSTEMS FOR YEARS. FINALLY, IT IS BEGINNING TO PENETRATE THE EMBEDDED SYSTEMS MARKET. VENDORS KEEN TO PROMOTE NEW MARKETS FOR THEIR SILICON NOW DRIVE INVESTMENT IN THE OS. SO, HOW EASY IS IT TO GET STARTED, AND HOW USEFUL IS IT?

Embedded Linux steals design wins

At a design review meeting five short years ago, my suggestion to consider Linux for an embedded system's user interface and supervisory control met with a lukewarm response. Objections ranged from Linux' inability to respond to real-time events, through concerns about software support within the open source community, to plain fear of the unknown—and the group eventually chose Windows CE. Back then, it's fair to say that no derivatives of mainstream operating systems properly suited embedded development, with the result that most designs employed proprietary solutions such as QNX Neutrino or Wind River's VxWorks. Meantime, the mainstream commercial and open-source communities have been busy adopting their core software architectures to suit small-footprint embedded systems—with the result, for example, that at last February's 3GSM World Congress, industry giants such as Infineon, Philips, Samsung, and STMicroelectronics announced cell phone products that use Linux. Elsewhere, Freescale's software arm Metrowerks offers automotive-strength Linux, primarily for in-car entertainment applications. And just last month, Xilinx announced a Linux port for its freely downloadable Webpack FPGA development environment, signalling the OS's increasing acceptance within the engineering fraternity. So just how useful is Linux for embedded development and—equally importantly—how easy is it to get started?

First off, it's essential to differentiate between "hard" real-time operating systems that guarantee no missing deadlines and where timelines are exactly known, and "soft" real-time systems that, on average, perform tasks within a specified timeframe. Environments such as QNX, VxWorks—and a host of others, such as the Realogy suite from LiveDevices and Volcano's automotive networking series—tackle mission-critical applications where fully deterministic response is essential (**Reference 1**). But as *EDN*'s Warren Webb recently reported, there are a number of commercial Linux products that target real-time needs, such as BlueCat Linux from LynuxWorks that complements its LynxOS, and the forthcoming Red Hat Embedded Linux, a cooperative effort from Linux stalwarts Red Hat and the biggest commercial RTOS vendor—Wind River—that harmonises Linux and VxWorks (**Reference 2**). Proof that these developments mean business comes from recent announcements such as General Dynamics Advanced Information Systems' selection of LynuxWorks' LynxOS-178 safety-critical RTOS for the US Army's future combat systems (FCS) infrastructure.

While such systems are Linux–compatible, they inevitably employ proprietary software to fulfil customer needs. For example, MontaVista's modifications to the Linux kernel for its carrier-grade product add numerous pre-emption points to improve worst-case latency from >100 msec to <1 msec to suit use within telecoms infrastructures. Alternatively, RTLinuxPro from FSMLabs runs Linux as an idle thread under supervision from a hard real-time OS core that guarantees worst-case interrupt latencies of 13 μsec on a typical x86 platform, while permitting seamless access to the Linux environment. European interest comes from Italian software

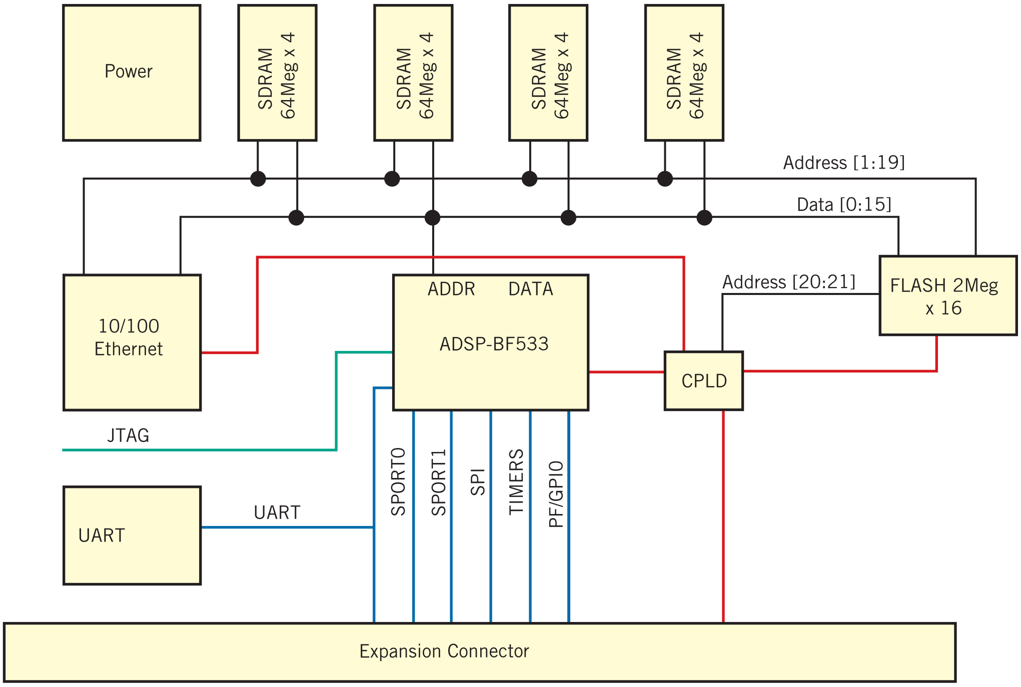


Figure 1

The Stamp board from Analog Devices enables Linux development via a 500 MHz Blackfin microcontroller and 128 Mbytes of SDRAM.

developers Koan, whose Klinux uses the real-time application interface (RTAI) kernel extensions from the RTAI Project at the Polytechnic of Milan's aerospace engineering department to furnish hard real-time response. Available under royalty-free open-source conditions, Koan specialises in porting its system to processors including x86, ARM, PowerPC, and Xscale architectures, principally for use within industrial controls.

Back in the consumer space, the spate of infotainment and telephony application announcements signals a trend for far more widespread embedded Linux adoption within devices that impose less severe demands on mission-critical sturdiness and real-time response. Of course, Linux already has an enviable reputation for reliability and security that helps explain its recent choice for a joint venture between mobile phone operator Orange and STMicroelectronics, where the OS runs a secure electronic cell phone payment application on STM's ARM-based Nomadik processor. But Linux' traditional response-time tardiness has been a barrier for many applications, which has been due to the original kernel's non-pre-emptive design—where running tasks suspend interrupts—along with the use of a fairness algorithm for its scheduler that guarantees all tasks a slice of processor time, regardless of priority considerations. This kernel also depends on page-swaps between tasks, which makes it impossible to accurately predict timing in systems that must accommodate asynchronous events. It also requires a memory-management unit (MMU) for memory protection and to facilitate multitasking.

Today's version 2.6 of the kernel includes enhancements to reduce latency along with many embedded-friendly features, including the option to build small-footprint kernels for devices that don't require user interfaces. Included in kernel releases from the interim version 2.5, the heart of the system is the so-called O(1) scheduler that adds kernel pre-emption, task priorities, and a dedicated time slicer. Mark Spencer, president of Linux-based telephony-systems builder Digium, reports that the OS's performance is fine for soft real-time applications without additional kernel modifications: "Most modern PC platforms can handle as many as 1,000 hardware interrupts/sec, which is adequate for the time-division-multiplex busses at the core of Digium's Asterisk system," he says. He considers that one of Linux' key advantages is its ability to scale from tiny embedded platforms to supercomputers and parallels its development with that of the PC: "In the beginning, PCs just ran word processors and spreadsheets, but they soon began to displace minicomputers and dominate the computing landscape. Linux is becoming similarly pervasive, with products now appearing in hugely diverse application areas." Spencer observes that Digium's continuous development of its open-source Asterisk offering, which the company can also license to third parties for proprietary software development, typifies the Linux commercial fraternity's business model: "We've just launched Asterisk Business Edition and are working on more PBX infrastructure hardware, including an echo cancellation card," he says.

Another hardware consideration is Linux' requirement for 32-bit machines. There is a 16-bit project that targets 8086-type architectures at http://elks.sourceforge.net, but with the availability of 32-bit chips such as Atmel's AT91SAM7S family of ARM7-based Flash microcontrollers costing around $3 in volume, it's pointless pursuing 16 bits for anything other than historical curiosity. Of course, Linux' own history of development on 386 machines means that it's easy to port into the 386 embedded space. But there's also widespread support for ARM and PowerPC architectures, along with various projects that include target machines such as MIPS and 32-bit Renesas Super-H devices that include MMUs. The uClinux project (www.uclinux.org) makes it possible to run Linux on machines that lack MMUs, which is significant for the embedded space—where there's often little demand for multitasking. Unlike many other Linux projects that currently appear fairly dormant, uClinux is definitely alive and well. In fact, kernel version 2.6 includes features from this project, with the result that it's now possible to run the OS on MMU-less chips, such as Analog Device's Blackfin, the ARM7TDMI, Freescale's 68k/ColdFire and QUICC products, Intel's i960, and NEC's V850E. If you must have 16 bits, there's even an active project for the H8S/2100 from Renesas (see http://sourceforge.net/projects/h8-uclinux/).

LOW-COST DEVELOPMENT SYSTEMS

All of this may sound just great. However, the available choices make getting started with Linux a nontrivial exercise. As all embedded development requires a host machine, it's next-to-essential to have a desktop Linux installation (see **sidebar** "Is Linux fit for prime time?"). If this choice is complex, choosing an embedded platform is even more difficult. For instance, a search for embedded English-language Linux at www.linux.org returns no less than 27 distributions for all platforms. As for the desktop environment, it's perfectly feasible—and within the spirit of the Linux community—to download freely available files and compile your own installation. Engineers with less time and more immediate focus are more likely to benefit from ready-made development platforms, and again there's a wide choice. For example, Koan offers a start-up kit that includes the software, documentation, and the company's development environment/toolbox from €299. At the opposite end of the spectrum, Arcturus Networks—the professional developer of uClinux—offers a reference platform for residential gateways and routers, along with complete development kits for Atmel's AT91-ARM7TDMI as well as for Freescale's ColdFire and DragonBall architectures at prices that span $495 to $1295.

For engineers interested in applying Linux for applications as diverse as embedded controls and infotainment consoles, the recently released ADSP-BF533-Stamp development kit from Analog Devices (ADI) makes interesting news. For less than $200, the product comprises a target board, a CDROM that contains the uClinux software, and a universal-input power supply—just add a standard straight-through serial cable and a host PC and the system is complete. The board accommodates a single ADSP-BF533 Blackfin 500 MHz processor along with a seemingly massive 128 Mbytes of SDRAM that's arranged in four banks of 16×4 (**Figure 1**). This memory complement gives software writers sufficient latitude for application development at a time when multimedia phones are demanding as much memory as some PCs. For example, Samsung has just announced a 4 Gbit flash memory that packs four 1-Gbit die into an 11×13 mm package. The 1.8V device sustains data reads at 108 Mbytes/sec—four times faster than conventional NAND Flash devices—and writes at 10 Mbytes/sec, which is over 60 times faster than standard NOR Flash. The company plans mass production starting this July.

The Stamp board accommodates 4 Mbytes of Flash together with a CPLD that expands the processor's asynchronous memory space to suit this maximum amount of Flash. Various peripherals, such as a 10/100 Mbit/sec Ethernet port, appear within the processor's asynchronous memory-map space. Other hardware highlights include three pushbutton switches and LED indicators that the CPLD can map into GPIO space or disconnect, and a multilevel switch-mode power supply that displays the capabilities of ADI's ADP3025 controller chip. Two headers that allow access to the Blackfin's and the CPLD's JTAG interfaces complement the normal range of headers that access the microcontroller's peripherals (**Figure 2**). These peripherals include two serial ports, an SPI port, a parallel-port/GPIO interface, three timers, a real-time clock, and an infrared device interface. These features fit onto a four-layer board that measures 178×127 mm. An ADC daughter-board is available, and hardware now under development (for which schematics are already freely available) includes an audio codec card and a pair of video encoder/decoder cards.

The CDROM contains the system's documentation, including schematics and CPLD files, and various software components. A 130-page user manual that includes sections from Analog Devices and Arcturus Networks introduces uClinux and documents the Stamp board. It includes some tutorial material but is currently in the pre-beta stage (release 0.5.4); for updates, check the user group pages at http://blackfin.uclinux.org, which is also the portal for support issues. The CDROM's software content includes pre-built kernel images and the source code, the toolchain ports for the Blackfin, a free distribution of JTAG tools, the U-Boot loader, and a Blackfin port of the Cygwin environment. A single-sheet quick-start guide satisfies users anxious to make something happen. The first task is to establish PC-to-board communications using, for instance, HyperTerminal. Follow the few instructions and the board powers up, runs through some memory check routines, and defaults to boot the kernel from onboard Flash. The uClinux command shell screen duly appears within the HyperTerminal window.

To avoid the necessity of installing Linux on the host—in the case of this evaluation, a Windows XP Professional machine running SP2 on an NTFS-

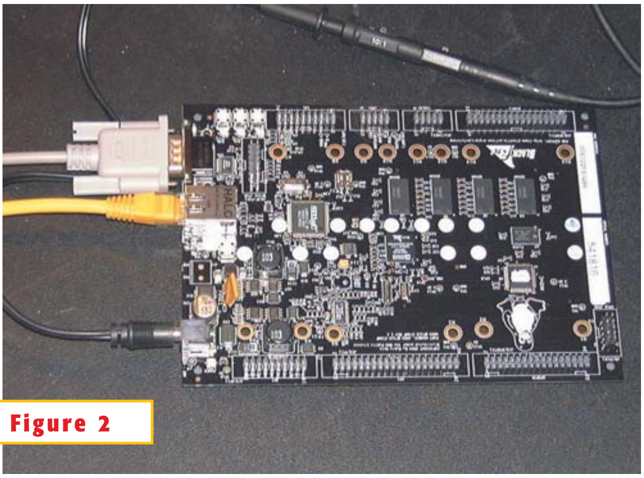


Figure 2

Headers provide access to all peripherals while offering JTAG connectivity to programme the ADI Stamp board's Flash.

IS LINUX FIT FOR PRIME TIME?

As the main text asserts, embedded Linux developers require a desktop instance. And while Linux gurus promote their favourite installations, newcomers often wonder what's different about packages from mainstream vendors such as Mandriva (formerly Mandrake), Red Hat, Slackware, and SuSE—especially as it's possible to download and build an installation from, for instance, Debian (www.debian.org) or Fedora (www.fedoraproject.org) for free. Interested users should also check out the lesser-known Gentoo (www.gentoo.org), a Linux flavour that's expressly tuned for scalability. But first-time users need determination to build their own systems from Web resources, whereas for typically less than €100, commercial distributions provide a package that's guaranteed to install relatively quickly, together with support documentation that's often quite good. All such distributions contain similar core components and mainly differ in the additional software that comes in the bundle—such as graphical set-up tools that augment traditional command-line configuration commands. But it's worth checking which kernel version the distribution includes, although the latest release isn't necessarily the most stable—hence commercial users typically recommend using one that's a few issues old, unless it's known to have serious flaws.

Another point that's worth considering is the distribution's ability to detect hardware. Although the major vendors have improved their products' installation routines significantly in recent years, experience suggests using a PC that's as standard as possible—not so long ago, USB support was flaky, and it's only in the last month that an everyday Windows-PC utility such as CDROM/DVD burning software from Nero (Germany's world leaders in PC optical-writer software) has become available. Notebook-PC support is supposedly available, too, but I haven't yet dared to test this on a brand-new Dell Inspiron that (astoundingly) lacks re-install disks for its Windows XP and driver software. Historically, Linux notebook support has been inadequate at best, hence the test PC in this evaluation is a conventional desktop that has a MicroStar International motherboard with a 2.4 GHz Pentium-4, 512 Mbytes of RAM, and the AMI bios. Peripherals comprise two IDE-interface 80-Gbyte hard disks (serial ATA apparently still suffers from immature support under Linux), a Sony DVD/CDROM reader and companion DVD writer, an Acer multimedia USB keyboard, a Microsoft PS/2 mouse, and an NVidia GeForce4 MX440 graphics adapter driving an Iiyama Vision Master Pro monitor. The resident OS is Windows XP Pro.

The ADI Stamp documentation (see main text) shows SuSE screens, reflecting the fact that this distribution from Novell's German subsidiary is Europe's leading implementation. As the release of the much-awaited version 9.3 of SuSE Professional arrived just in time for this edition of EDNE, it's the natural choice for a test drive. The package comprises two DVDs and five CD-ROMs that replicate the installation and application binaries and source code; a 280-page user guide; and a 680-page administration guide that contains installation and system configuration information. Application software includes a range of database, graphics, Internet/email, mobility, multimedia, office, and security applications, as well as the KDE and Gnome desktop environments—and masses of development tools.

Caution dictates backing up the Windows installation before creating a spare 10-Gbyte area on the second hard drive. Install options include the *grub* boot-loader that allows the system to start Linux or Windows, the KDE desktop, and all Linux development tools, making the installation size about 4.7 Gbytes. The only step necessary to install from DVD was to enable bios legacy USB support to enable the installer to recognise the keyboard. Checking the hardware tool confirms that the installer correctly identifies the system's components. The installer automatically found the LAN and established an Internet connection via the network's ADSL router using DHCP (dynamic host-configuration protocol), connecting into Novell and updating SuSE with recent patches.

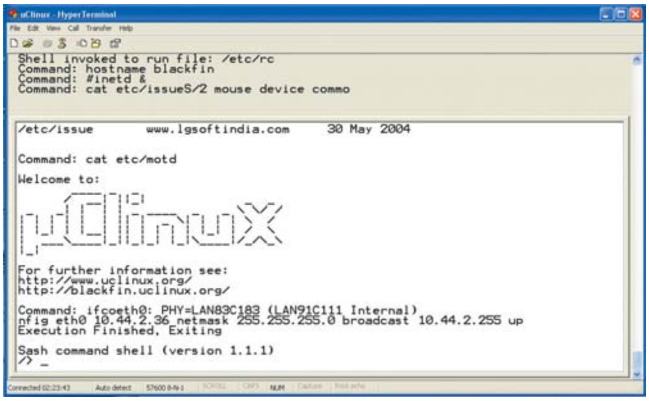
But, confirming the experience of nearly 10 years of working with Linux, this is where immediate reward stops. Certainly, if you just want to run the integral OpenOffice development of the excellent StarOffice suite that Sun Microsystems hosts, the Linux desktop is a comfortable place. But try to connect a Linux box to a WinNT4 server network and it's still necessary to grapple with the often-arcane internals of Samba, the Unix-to-DOS/OS2/Windows LAN-bridge software. Plus, simply clicking on the network icon to browse returns the unhelpful error message, "lisa daemon not installed or running". Resolving this issue is neither trivial nor covered in the documentation, thus requiring—as is often the case—a Web search to find help. (Sadly, "The Linux Network"—my favourite resource for Samba and Linux networks in general—is currently out-of-print, but still available: see **Reference A.**)

There's also some evidence of immature application-level software, as the first attempt to build a search index within KDE's help system crashed the application (but not the system). In many ways, this experience reflects an impression of the Linux/Windows divide—experience here with kernel-level development on a 2.4 installation confirms that it's very hard to crash or damage Linux. But it wasn't until Windows 2000 Professional that Microsoft produced an OS that rivals Linux's stability, with Win 2K Pro demonstrating rock-solid performance in over three years of intensive, every-day use. Meantime, it's easy to feel that the Linux camp has made little progress in easing its desktop into the mainstream beyond implementing numerous "me-too" features from its rival, with the formerly clean lines of KDE increasingly adopting XP's look-and-feel. But power users will appreciate the ability to tailor and continuously update their Linux systems without being forced into the wholesale changes that have made XP unpopular in some quarters. Crucially, developers used to wrestling with system issues won't encounter any insurmountable difficulties, and will soon come to love the Linux environment—especially with its open access to the technology—but it still isn't a breeze.

Reference

A. Butzen, Fred & C. Hilton, "The Linux network," IDG Books, ISBN 1-555828-589-X. Out of print but still available—see www.amazon.com.

format drive—users can install Cygwin, a Linux-like environment for PCs (www.cygwin.com). As its home page forewarns, Cygwin is not a way to run native Linux applications on Windows, or to make Windows aware of Unix functions: "You have to rebuild your application from source if you want to get it running on Windows." But this restriction is irrelevant in the embedded context, where every change to the target requires recompilation. The Cygwin instructions that accompany the kit also estimate that this port is around three times slower than a native Linux version. It's no lightweight installation either, occupying almost 1 Gbyte of disk space. But the Blackfin port of Cygwin includes the toolchain and the uClinux kernel, which avoids the requirement to download and build these components on a Linux host. This approach short-circuits tasks that first-time users may find intimidating, although the main user manual provides step-by-step instructions that show how to build the toolchain on a Linux box.



```
Shell invoked to run file: /etc/rc
Command: hostname blackfin
Command: #inetd &
Command: cat etc/issueS/2 mouse device commo

/etc/issue       www.lgsoftindia.com     30 May 2004

Command: cat etc/motd
Welcome to:

For further information see:
http://www.uclinux.org/
http://blackfin.uclinux.org/

Command: ifcoeth0: PHY=LAN83C183 (LAN91C111 Internal)
nfig eth0 10.44.2.36 netmask 255.255.255.0 broadcast 10.44.2.255 up
Execution Finished, Exiting

Sash command shell (version 1.1.1)
/> _
```

Figure 3 Cygwin and HyperTerminal permit uClinux development and communications on a Windows-PC host.

Running Cygwin's setup decompresses and installs a virgin system. The set-up procedure also installs the freeware AnyEdit text editor that can convert between Windows Cr-Lf (carriage-return/linefeed) and Linux Lf end-of-line terminators (other first-time-user issues to beware include Linux' use of the forward-slash rather than Window's backslash for path direction). Clicking on the Cygwin uClinux desktop icon then opens the system's command-line interface window that's based on the Linux-standard "bash" shell, the Bourne-again-shell replacement for the shell that's used throughout the Unix world. Next, the Cygwin instructions walk the user through the make process that tailors the environment to the target board. This step appears under the configuration screen's "processor type and features" submenu, and defaults to the correct options for the Stamp target. For first-time use, ignore the temptation to tamper with these screens—that is, simply select Save & Exit when the SnapGear configuration screen appears— then invoke the make command. This triggers a lengthy script that builds the system and configures the toolchain for the Stamp board, taking upwards of 15 minutes on a 2.4 GHz Pentium-4. But don't forget to reserve investigating the SnapGear screen options for later, as they allow quick kernel rebuilds with alternative parameters.

The process concludes by depositing a file that's simply called *linux* (not *linux.dxe*) in the *.../uClinux-dist/linux-2.6.x/* directory. This file is an executable-&-linkable format (ELF) uClinux kernel image that users can download via the U-Boot loader and run from the Stamp's SDRAM. The Cygwin installation documentation doesn't explicitly tell you so, but at this point, the system is ready for its first use with the Stamp target. It's then best to jump to U-Boot's HyperTerminal section in the user manual, which describes how to download the newly compiled kernel image and run it. This is the least-effort way for new users to ensure that the system is working, but it takes about 1.5 hours to download the 6.3 Mbyte file over the test PC's 57,600 kbps Kermit-protocol serial link.

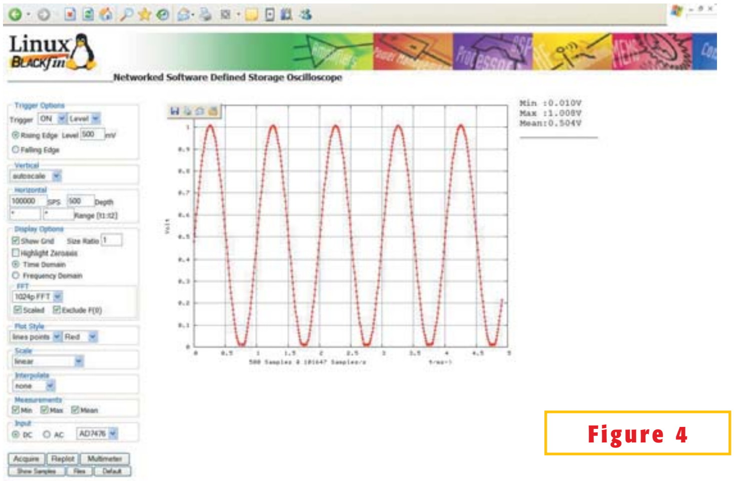
AT A GLANCE

- ▷ Embedded Linux scores with major telephony design wins
- ▷ Semiconductor vendors promote Linux to sell silicon
- ▷ Entry-level hardware enables serious work for <$200
- ▷ Open-source offers freedom but raises IP ownership issues
- ▷ Desktop OS adopts XP look-&-feel

When the HyperTerminal screen returns, typing *bootelf 0x1000000* should reward patience with a new instance of uClinux that's running out of SDRAM, confirming that the system installation is healthy (**Figure 3**).

Because the serial link is so slow, it's imperative to get the Ethernet communications working. This requires connection into a LAN via a standard Ethernet RJ-45 cable, or via a crossover cable directly into the host PC. Again, the documentation does not explicitly state that the Ethernet connection doesn't come alive until you invoke a U-Boot command such as *tftpboot*—at which point the on-board Ethernet activity LEDs will light, as will monitor LEDs in the LAN's switch. Although obvious in retrospect, this departure from standard LAN practice caused some consternation here that resulted in swapping cables, trying Ethernet crossover wiring, and then 'scoping the Ethernet controller's chip-select line while ping-ing the network, before the clue finally sunk in. Similarly, ping only works from the target to the server, timing out in the opposite direction as there's no process implicitly running on the target to complete the handshake. Linux veterans invariably consider such issues as a "newbie's rite-of-passage", hence it's imperative to be prepared to dig-and-delve to get results—and in reality, there are a several full-blown commercial products that are no less arcane.

The LAN connection requires a TFTP (trivial-file-transfer-protocol) server running on the host, such as the free-for-personal-use TFTP Turbo Windows-compatible software that the manual describes. Alternative solutions include Free TFTP Server from network-management specialists SolarWinds.Net, which is available as freeware. Correct installation and setup cuts the download time for 6,491,055 bytes to about 18 seconds on the test PC, when running the *bootelf* command loads and runs the kernel image. In case of doubt, check the header information for the timestamp of the image that's now running—but don't save any environment variables using U-Boot's *save* command until certain



Networked Software Defined Storage Oscilloscope

Trigger Options
Trigger ON Level
● Rising Edge Level 500 mV
○ Falling Edge

Vertical
autoscale

Horizontal
100000 SPS 500 Depth
* * Range [t1:t2]

Display Options
☑ Show Grid Size Ratio 1
□ Highlight Zeroaxis
● Time Domain
○ Frequency Domain

FFT
1024p FFT
☑ Scaled ☑ Exclude F(0)

Plot Style
lines points Red

Scale
linear

Interpolate
none

Measurements
☑ Min ☑ Max ☑ Mean

Input
● DC ○ AC AD7476

```
Min :0.010V
Max :1.008V
Mean:0.504V
```

Figure 4

The network software-defined oscilloscope project demonstrates a remotely controlled Web instrument running under uClinux.

that they work correctly, as this step flashes system memory. Recovering from a bad Flash situation requires the user to reprogramme the memory via the Stamp board's JTAG interface employing a low-cost parallel-port wiggler. The hardware schematics and software files appear at the ucLinux/Blackfin portal, but details of the process are currently missing from the user manual, signalling the need for care here.

At this point, the user has a fully working uClinux development system. It's then possible to recompile the kernel to reduce footprint from the command line or by using the SnapGear graphical interface (a process that the user manual describes well). Michael Hennerich, European DSP systems and applications engineer at ADI Munich, Germany—and a leading architect of the uClinux/Blackfin project—notes that the default root filesystem size is around 4 Mbytes, with the default kernel configuration occupying about another 1 Mbyte. Users can adjust these footprints to suit their applications. Hennerich also points to the U-Boot loader's facility for compressing and decompressing Flash-memory kernel images that can achieve as much as a five-fold Flash memory saving at the cost of a slight increase in boot time, and cites the Blackfin's dynamic power management abilities that further optimise the processor for low-power, small-footprint systems. While kernel compilation permits static changes to clock-frequency parameters that default to 497 MHz core and 124 MHz system clocks, a real application can dynamically vary the clock frequency and core voltage to conserve power in portable electronics. Hennerich says that it's premature to release formal benchmark results, but says that his own tests with MP3 decoding reveal 5% processor loading, hence there's considerable latitude for power management in portables.

Hennerich also directs developers' attention to the BusyBox programme that crams many familiar Unix facilities into a small-footprint executable. Requiring only a kernel image and the */dev* and */etc* filesystems, BusyBox—the "Swiss army knife of embedded Linux"—furnishes the key GNU toolchain's core utilities that any embedded Linux system needs. Supplied as source code, BusyBox is processor-agnostic and allows users to include only those functions that the application requires. Contributors to its code base include Linus Torvalds, the creator of the Linux OS (see www.busybox.net).

KERNEL MANIPULATION FOR I/O

Because Linux handles all I/O via device drivers, getting the OS to do something useful requires writing and debugging kernel-level code. The current Stamp documentation includes several pointers to help users accomplish this task, as well as many helpful sections on issues such as using the *make* command and the *gdb* Linux debugger. But the documentation package currently falls short of describing key issues that include application and driver development. ADI advises that the documentation is under review and that new material will appear very soon. ADI's Hennerich points to the *pflag* driver that appears on the portal as a simple example of an application that toggles one of the Stamp board's LEDs, terminating when the user presses the matching pushbutton. This example and many more reside within the portal's CVS (concurrent-versions-system) repositories (**Reference 3**). One work of particular interest is the network oscilloscope project that uses the ADC daughter card and open-source driver software to demonstrate how to set up a Web server that remotely controls the instrument, returning jpeg-format traces to the host in either time or frequency domains (**Figure 4**). For this and any other serious development work, Hennerich stresses the need for a proper Linux host: "Cygwin is suitable for test drives, but serious developers should use a native Linux distribution." He adds that he's working on another Cygwin release that's intended to show an I/O port manipulation, as well as a replacement called coLinux that he recommends for future use on Windows systems. coLinux runs a real Linux kernel—complete with its file systems and case sensitivities—on a Windows machine, and will load native Linux applications. Hennerich emphasises that work is ongoing on support documentation, and urges users to regularly check the Blackfin/uClinux portal for updates to this and other packages.

Other available quick-start resources include a series of features in EDN's sister publication Test & Measurement World, which also lists further resources, such as several invaluable books from the O'Reilly Media series (**Reference 4**). A recent addition to O'Reilly's catalogue is Karim Yaghmour's "Building embedded Linux systems", which painlessly combines a great deal of information that would otherwise take endless Net searches to acquire. The publisher also offers the third edition of Alessandro Rubini's now classic "Linux device drivers" handbook that's updated to cover the 2.6.x kernel. But if the thought of navigating the open-source community doesn't appeal, there are numerous alternatives becoming available that still provide cost-effective entry-level opportunities. For example, ADI has uClinux ports for the BF531, 533, and 535 Blackfins that suit its EZ-KIT hardware and VisualDSP++ development environment. It's also possible to marry the Stamp kit to VisualDSP++, for which a 90-day evaluation license is available.

Michael O'Donnell, manager of core technologies marketing for Freescale's software subsidiary Metrowerks, summarises Linux penetration into the embedded space by observing that back in 2000, there was considerable interest but little adoption. Now, his customers express great interest in real applications such as automotive infotainment. O'Donnell observes that the entire Linux business model is changing, and that today's heavy investors are semiconductor makers keen to establish new markets for their silicon: "As a result, the prime users are no longer Linux gurus, and carry over their expectations from commercial embedded development environments," he says. It's therefore imperative that software vendors make embedded Linux easy to approach and use. He characterises a typical development team for a project of any size comprising a couple each of kernel- and driver-software writers, and perhaps as many as 20 application-software writers. For this reason, it's essential to have an integrated development environment that allows engineers to seamlessly move from one project phase to another: "Customers are looking for hardware and board-support packages that span evaluation and feasibility studies to application-software and kernel-level debugging in complex systems." Metrewerks accordingly offers a range of products based on its CodeWarrior toolchain that suit Freescale's ARM, ColdFire, and PowerPC architectures. Those products include the QUICCstart evaluation packages that bundle a development board, the CodeWarrior integrated-development-environment, a Linux board-support package, as well as the WireTAP run-control tool for the PowerQUICC I, II, and III processors at prices ranging from $495 to $1095. O'Donnell also notes that recent ColdFire silicon includes an MMU, thereby facilitating a full Linux board-support package for this variant.

Commenting on issues that continue to concern customers, O'Donnell reports that intellectual property (IP) figures high on the list. A recent due-diligence search revealed around 50 styles of open-source licensing arrangements, leaving customers wondering how best to partition their precious IP from public-domain software: "Which licenses are included in the board-support package, and how do they interact with my IP?" is a question that he's frequently heard, especially from the traditionally conservative automotive segment. Metroworks released its automotive-grade Linux to widespread interest at Detroit's Convergence 2004 show. The technical lessons that the company learned in developing its product include the need for new versions to offer improved boot times, power management, and—even though not a hard real-time product, for which Metrowerks recommends its OSEKturbo RTOS—determinism. O'Donnell considers that stability is a key customer requirement, not only in the software where customers often choose mature kernels over the latest builds, but also in the overall business model that facilitates ongoing development from multiple suppliers. He concludes: "When Linux crosses the chasm and becomes more mainstream, the business model will mature to the embedded community's benefit."

You can reach Contributing Technical Editor David Marsh at forncett@ btinternet.com

Present indications from the stream of Linux-related announcements suggest that this maturity will come soon. For example, just as *EDNE* was closing for press, Qualcomm declared that it will support the Linux operating system on its mobile-station-modem MSM6550 chip sets. According to its press release, the company expects this move "to give handset makers additional design and development efficiencies for 3G smartphones and other mobile handsets." It notes that integrating Linux support into the chipsets eliminates the need for a separate coprocessor, and lowers design costs and complexity compared with the multiple-chip implementations that a third-party OS currently requires. Qualcomm also plans to extend Linux support to other chips in its enhanced multimedia platform family, including UMTS, HSDPA, and CDMA2000 derivatives.□

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, contact any of the following manufacturers directly, and please let them know you read about their products in *EDN*.

Analog Devices
www.analog.com

Arcturus Networks
www.arcturusnetworks.com

Digium
www.digium.com

Freescale Semiconductor
www.freescale.com

FSMLabs
www.fsmlabs.com

General Dynamics Advanced Information Systems
www.gd-ais.com

Infineon Technologies
www.infineon.com

Koan Software Engineering
www.koansoftware.com

LiveDevices
www.livedevices.com

LynuxWorks
www.lynuxworks.com

Mandriva
www.mandriva.com

Metrowerks
www.metrowerks.com

Microsoft
www.microsoft.com

MontaVista Software
www.mvista.com

O'Reilly Media
www.oreilly.com

Nero
www.nero.com

Philips
www.philips.com

QNX Software Systems
www.qnx.com

Qualcomm
www.qualcomm.com

RTAI Project (Polytechnic of Milan department of Aerospace Engineering)
www.rtai.org

Red Hat Software
www.redhat.com

Samsung
www.samsung.com

Slackware
www.slackware.com

SolarWinds.Net
www.solarwinds.net

Sun Microsystems
www.sun.com

Suse Linux
www.novell.com/linux/suse/

STMicroelectronics
www.st.com

Volcano Communications Technologies
www.volcanoautomotive.com

Wind River Systems
www.windriver.com

Xilinx
www.xilinx.com

REFERENCES

1. Marsh, David: "Automotive design sets RTOS cost and performance challenges," EDN Europe, September 1999, pg 32. www.edn-europe.com

2. Webb, Warren: "Embedded Linux nears realtime," EDN, September 16th 2004, pg 55. www.edn.com

3. Path to *pflags-test* example: http://cvs.blackfin.uclinux.org/cgi-bin/cvsweb.cgi/uClinux-dist/user/blkfin-test/pflags-test/?cvsroot=uclinux533

4. Marsh, David: "Linux applications cruise with new drivers," Test & Measurement World, April 15th 2000. http://www.reed-electronics.com/tmworld/article/CA187384 and "The ins and outs of Linux kernel device drivers," T&MW October 15th, 2000. http://www.reed-electronics.com/tmworld/article/CA187432