TOSHIBA AMERICA ELECTRONIC COMPONENTS

Motor driver IC cuts power consumption.

## SEMICONDUCTORS & ICs

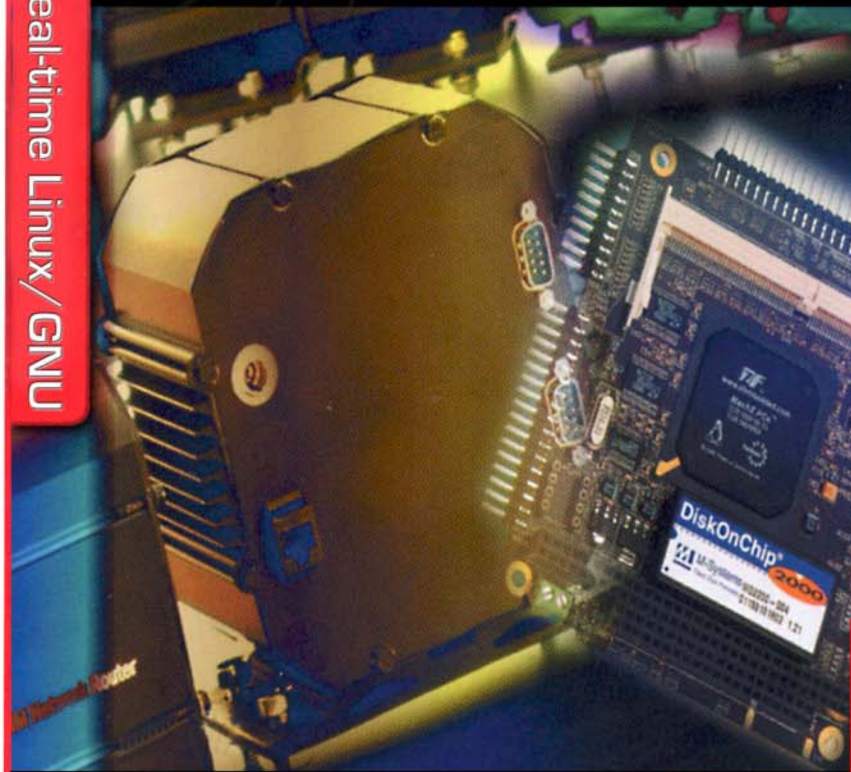
PRODUCT FEATURE
Page 23

# ep&t

JUNE/JULY
2001

ELECTRONIC PRODUCTS AND TECHNOLOGY

Real-time Linux/GNU



## REAL-TIME LINUX/GNU:

Free open-source embedded alternative.

PANASONIC

Lithium ion battery pack with safety circuit is for general purpose.

## BATTERIES & POWER SOURCES

TECHNOLOGY UPDATE
Page 50

WAGO

Modular system adds double deck terminal block styles.

## TERMINAL BLOCKS

NEW PRODUCT HIGHLIGHT
Page 64

Electronic Commerce
An EP&T Special Supplement
Pg 36

e p & t
ENGINEERING Design

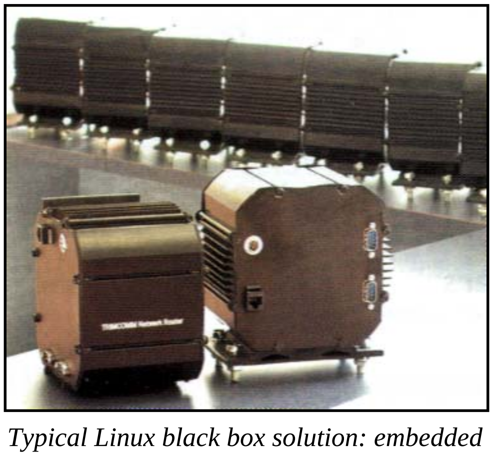
## Embedded real-time Linux/GNU represents free open-source alternative

*by Doug stead, president Tri-M Systems Inc, Coquitlam BC.*

Someone once said, “there is no such thing as a free lunch!” But according to the GNU organization (www.GNU.org) there is free software. This refers, of course, to Linux/GNU: users have the freedom to run, copy distribute, study, change, and improve software. This free software is all based on one precondition: free access to the source code.

Linus Torvalds and Richard Stallman can aptly be described as founding fathers of the open source code movement. These software gurus, legends in their own time, created what has become known as Linux. But, more accurately, it should be described as the GNU/Linux operating system (GNU is an acronym for Gee Not UNIX). Stallman's and Tolwald's contribution to our cyber-information age rivals that of George Boole, who in 1847 formulated the Boolean Algebra functions AND, OR, and NOT, on which all binary computer code is based, and that of Drs Bardeen, Brattain, and Shockley, who, in 1947 at Bell Labs, invented the transistor.

Stallman and Torvalds, together with many others developed from scratch what has evolved into a vastly popular OS platform. Their vision in creating Linux/GNU was to clone the venerable, but expensive, UNIX operating system. Further, this software was to be freely a distributable platform, to which thousands and thousands of programmers around the world have subsequently contributed to and expanded upon. Being that it is open source, these programmers are encouraged to view, use, and modify anything and everything to improve this OS.



*Typical Linux black box solution: embedded Linux/x86 router for hostile environments onboard large open-pit mining equipment. It gathers and packages data and passes it back and forth to a wireless communication system. It has to withstand constant shock and vibration in addition to noisy power source and the occasional spike or load dump, which can exceed 600 joule.*

Linux was originally developed to run on 80386 CPUS, the first true 32-bit x86-based PC. Today it has also been ported to run on Compaq Alpha AXP, Sun SPARC and Ultra-SPARC, Motorola 68000-family CPUS, PowerPC, ARM, Hitachi SuperH, MIPS, DEC VAX, Intel IA-65, and ether platforms. Quite a long journey for an operating system, which shows the great demand for this one while speaking volumes on its portability, sociability, and reliability.

GNU/Linux in its purest form refers specifically to the free-release kernels. It can also be correctly thought of as an entire operating system, which may include many inherent applications programs. From an embedded point of view, this kernel is likely the entire operating system code stored in flash and runs on an embedded X86-type module such as MZ104 PC/104 CPU module.

The GNU/Linux kernel controls hardware, managing files, and separates processes together with all other tasks needed to be managed in any particular embedded device. This is also the part that is free, providing you have the skill to build your own kernel, which will support your hardware, 1/0 and application to met the project's requirements.

If you would rather not build your own kernel, there are numerous commercial distributions on the market, such as RedHat, BlueCat, and REDsonic. These distributions offer many added enhancements to the standard free code and include many and various utilities and applications. They can make life a lot easier, with installation tools and graphical user interface (GUI), facilitating easier customizing of individual kernels to match the hardware and software requirements of your particular embedded project.

The latest open source kernel is version 2.4, which was released in January. It replaces the highly successful 2.2 kernel, of January 1999. OEMS, and other venders of critical service applications, such as drywalls, SCADA, robotics and communication devices, have for many years embraced GNU/Linux for its very high reliability.

## De-facto embedded standard

PC/104 and PC/104+ cards have become the de-facto standard for embedding x86 hardware. The MZ104 shown here is a typical 486/586 PC/104 single board PC, capable of booting Linux or a host of-other OSs from flash or rotating media. This 3.6x3.8" board draws less then 2 watts from a 5Vdc input, It provides OEMS with almost every feature you would expect from a desktop computer except video or LCD, to keep costs down, as very few embedded Linux solutions require a display screen. However if a GUI is required, the soon-to-be-released MZ104EV will do the trick or you can simply add a PC/104 video card on top.

The 2.4 release has focused primarily on improving performance en larger machines such as IBM'S S/390. However, it also expanded support in areas important to the embedded x86 community. The 2.4 kernel release new includes features such as universal serial Bus (USB) hardware support and support for file sizes larger than 2Gb. REDsonic (www.redsonic.com) has recently elevated the capabilities of the standard Linux kernel to another level by adding hard real-time support with its particular distribution. Embedded real-time systems seek to guarantee that the right operations, based en the right information are executed at the right time to meet the needs of a mission-critical application.

The key word here is “right”. Imagine an artificial intelligence embedded system applying the breaks to a train, after a collision. Clearly: in this example, the execution of the “right” operation is not enough; the time at which the operation is executed is equally important.

Currently, real-time systems are being used by stock trading, remote multimedia, national defense, aerospace, and intelligent-electronics medical and automotive systems. By integrating hard real-time into the Linux kernel, REDsonic offers the OEM or application developer the ability to develop mission-critical systems, ones that are reliable, predicable and responsive:

## Linux/GNU kernel evolution

The current Linux/GNU kernel has evolved today to include features one would expect to find only in expensive full-fledged Unix, Novel or Windows NT platforms:

- True multitasking.
- Virtual memory
- Shared libraries.
- Demand loading.
- Shared copy-on-write executables.
- Efficient memory management.
- X Windows.
- TCP/IP networking.

- Reliability: in real-time embedded systems, tasks share execution budgets and/or system resources, and are independent of each other. Should one task misbehave it should have no bearing on the execution of another task. In this way, one unreliable task should not jeopardize the overall stability of the system.
- Predictability: using a combination of scheduling methods to provide the desired predictable behavior. Time-driven scheduling ensures the precise time at which an operation is activated. Share-driven scheduling makes certain that every task receives a fixed percentage of system resources, while priority-driven scheduling guarantees that tasks with higher priority will always be executed before a lower priority task. Together these scheduling paradigms cover all scheduling possibilities thus increasing predictability.
- Responsiveness: this requires preemption points, which give higher priority tasks the ability to preempt lower priority tasks running in the kernel. This makes application more responsive as kernel blocking delays can be managed. Moreover, since each task can be restricted to its execution time budget, it will not delay other tasks for an extended time. These two properties make a real-time system much more responsive.

***Tri-M Systems Inc. (www.tri-m.com) and associated firm Tri-M Engineering engineer, design and market x86-based embedded boards and products, specializing in deployment in hostile and/or mobile environments.***

***Doug Stead, president,***
***Tri-M Systems Inc.***