PERSONAL MEDIA
CORP.
www.personal-media.co.jp

www.t-engine4u.com

Frequently Asked Questions about T-Engine Development Kit

T-Engine Development Kit FAQ

T-Engine®

Q 1 What is the flow of product development with T-Engine like?

Let's take the case of developing a new cellular phone. If you start developing the software for controlling hardware after the development of the hardware is completed, the overall development period would be too long. For shorter development period, the hardware and the software need to be developed simultaneously. In order to achieve this, before you start developing the software, you need to prepare a piece of prototype hardware (a board for feasibility assessment) that has similar configurations to the finished product (i.e., a board highly compatible with the software), regardless of the similarity in physical appearance. The hardware in this case, also called a breadboard, needs to be prepared quickly. It can be used as a general-purpose product, or with only minimal customizations and/or additional expansion boards or devices required. T-Engine meets these requirements. However, since T-Engine does not come with communication devices such as transmitter/receiver circuit or antenna, an existing cellular phone or a compact flash card with cellular phone functions will be required as the peripheral devices of T-Engine.

With T-Engine, you can also improve the efficiency of software development by using T-Kernel (the T-Engine's standard real-time operating system (OS)), sample middleware and device drivers that run on T-Engine. Meanwhile, you can start developing the hardware(finished product) to be mass-produced along with the software development.

Once you have finished the development of the hardware (finished product) and software that run on T-Engine, you can port the software on T-Engine including T-Kernel and the middleware you used, to the final hardware. You might need, then, to modify the software to comply with the final hardware if there is any detail difference between the hardware, such as memory maps. The more similar the specifications of the final hardware and T-Engine are, the easier the modifications will be.

This method might seem like a lot of work in that the software is developed on one piece of hardware (T-Engine) and migrate it to another (finished product). However, developing software on a common platform (T-Engine) has a lot of benefits: You can use your existing middleware, and stockpile your own software to be developed and development know-how for reuse. On the other hand, software development on the final hardware would lack the usefulness of stockpiling as such software tends to be customized for specific hardware. The utilization of a breadboard or T-Engine is thus very useful for shorter development period and higher efficiency.

As in this example, if the T-Engine board mentioned above is not used as the hardware of the finished product (cellular phone), instead, T-Kernel or some middleware on T-Engine is embedded in the product, you need to conclude separate license agreements with the T-Kernel manufacturer and relevant middleware developers (* See [Q. 13]). Please contact Personal Media Corporation or the relevant middleware makers for more details.

Q 2 What is the difference between T-Engine and µT-Engine?

The T-Engine project defines multiple hardware specifications according to the size and purpose of the target equipment in which computers are embedded: The "standard T-Engine" is for equipment such as personal digital assistants with comparatively advanced user interface like an LCD or a touch panel. The "µT-Engine" is for equipment like home appliances and instrumentation with comparatively few user interface features. Each T-engine standardizes the platform size and connector positions, as well as the basic hardware configurations.

The "standard T-Engine" defines the size of its CPU board as 75x120 mm (2.953 x 4.724 inch) and the "µT-Engine" specifies it as 60x85 mm (2.362 x 3.346 inch).

These T-engine platforms have different peripheral devices: the standard T-Engine has a PC-card slot and a USB port, whereas the µT-Engine has a compact flash (CF) card slot and an MMC card slot.

Both T-Engine boards have a serial port and an eTRON-card interface.

Please see the T-Engine Solution Website (http://www.t-engine4u.com/en/) and "TRONWARE" (TRON/T-Engine bimonthly magazine) for more information.

Q 3 What is the difference between ITRON (µITRON) and T-Kernel?

ITRON, with its loose standardization policy, has been widely used including in 8-bit CPUs. The TRON Project standardized the application programming interface (API) for the basic functions of the real-time OS. However, the TRON Project did not specify any rules or guidelines for the hardware for execution, device drivers, development environment, etc., which were left to the makers of ITRON-related products.

However, after considering the distribution and compatibility of middleware and device drivers, we decided that it is better to provide a set of standard specifications and guidelines. The T-Engine project, for this reason, widens the scope of standardization, and adopts a rather "strong standardization" policy.

Therefore, T-Kernel, the standard real-time OS for T-Engine, has functions that enable the distribution of middleware on top of the existing basic ITRON real-time OS functions. For example, the standard ITRON functions do not cover device management as opposed to T-Kernel that does, making T-Kernel possible to develop middleware and applications independent of device drivers. T-Kernel also has subsystem functions for building larger systems (virtual memory, etc.).

Provided, T-Kernel and µITRON 3.0 or µITRON 4.0 share common basic real-time OS functions such as tasks, semaphores, eventflags, messages, and rendezvous. Since T-Kernel is developed based on our twenty years of experience and development know-how with ITRON, any technician with previous ITRON knowledge should be able to adapt to T-Kernel easily.

Q 4 Could you please show me some specific examples of T-Kernel's APIs (system calls/service calls)?

T-Kernel has the following functions and APIs. The T-Engine development kit includes the T-Kernel specification handbook.

- Task control
- Task synchronization and communication
- Memory management
- Exception/interrupt control
- Time management
- Subsystem management

Table 1 T-Kernel/OS Functions

- System memory management
- Address space management
- Device management
- Interrupt management
- I/O port access support
- Power-saveing functions
- System configuration information management

Table 2 T-Kernel/SM Functions

tk_cre_tsk	**Create task**
tk_del_tsk	**Delete task**
tk_sta_tsk	**Start task**
tk_ext_tsk	**Exit task**
tk_exd_tsk	**Exit and delete task**
tk_ter_tsk	**Terminate task**

Table 3 Example of T-Kernel service call (part of task control functions)

Q 5 What programming languages can be used for development?

The T-Engine development kits come with a GNU development environment for the target CPU, enabling use of C language (gcc) for development. The GNU development environment runs on a PC-Linux machine.

It is also possible to use C++ and a development environment in Windows (Cygwin) with all the T-Engine/µT-Engine Development kits, except µT-Engine/M32104.

In addition, the PMC T-Shell (* See [Q.12]), which is an optional collection of middleware products, includes the Microscript visual language. Using this package, GUI-based operation screens, demonstrations, presentations and other visuals can be developed easily. And since Microscript programs run without modification in Chokanji 4, a PC operating system, a PC with its large screen can be used to carry out efficient design and development of user interface graphics.

Q 6 What debugging environments are available? Is debugging using JTAG or ICE possible?

The T-Engine development kit provides gdb, the source-debugger that GNU supports, making it possible to cross-debug via a serial cable.

The GUI-based integrated development environments and debuggers using JTAG or ICE are also available as options or products by other manufacturers. Please contact us for the latest information on T-Engine.

Q 7 Is development in Windows possible?

The T-Engine project defines a development environment reference model aimed at facilitating source code recompiling for other CPU architectures, and distribution of middleware at the object code level. This is a GNU environment. The actual development environment does not have to be GNU, but interworkability with the GNU environment and the ability to link to object code created in GNU are required of any development environment used.

The T-Engine development kits come with a GNU development environment running in Linux. If only the development kit is used, essentially development is carried out in Linux. In order to meet demands for development in Windows, we have made available a Windows version of the T-Engine development environment, using Cygwin, for all the T-Engine/µT-Engine Development kits, except µT-Engine/M32104.

We are planning to enable developers to use "Eclipse", the open source and widely used Integrated Development* Environment (IDE) all over the world, to develop software on T-Engine.

Q 8 Are the applications or middleware to run on T-Engine executed after linking it with T-Kernely?

Applications (including middleware) do not have to be linked with T-Kernel. Applications and T-Kernel belong to separate program modules. A T-Kernel system call is called from an application through a trap command (a software interrupt command, equivalent to an INT command of x86). This is similar to issuing a system call of the OS from an application when using an OS of personal computers(Windows, Chokanji, etc.) or UNIX.

The applications you develop will be executed after downloading it, via a serial cable, etc., to T-Engine from the machine used for the development (PC-Linux machine). As T-Kernel is written in the flash memory in binary format, you do not need to download T-Kernel every time, except for special cases such as upgrading the OS.

Q 9 What is the real-time performance of T-Kernel like?

As T-Kernel is a high-performance real-time operating system based on our 20 years of experience and development know-how with ITRON, its real-time performance is one of the best.

The following is an example of its performance data on the T-Engine/SH7727 development kit. Please note that the actual performance may vary according to factors such as clock frequency, number of interrupt handlers, and priority allocation:

* Maximum time to disable interrupts: About 2 microseconds (with a 140 MHz CPU); About 3 microseconds (with a 96 MHz CPU)

* Dispatch time after waking up the waiting task by wup_tsk(): About 7 microseconds (with a 140MHz CPU); About 10 microseconds (with a 96MHz CPU)

Q 10 What is the necessary memory size for T-Kernel?

It depends on the type of CPU, system configuration, and the number of objects such as tasks, but the size of the program is about 70KB, and that of data is at least around 8KB.

Q 11 What kind of devices can be used for T-Engine screen output?

The standard T-Engine supports an LCD panel with 320-by-240-dot touch panel. Note that the LCD panel (LCD board) is optional with some T-Engine development kit models. Be sure to confirm this at the time of purchase.

In addition, although these are not included in the T-Engine standard specifications, there are T-Engine models supporting connection to a CRT display used with personal computers, and μT-Engine models that connect to an expansion board for VGA output. The brochures or other documentation for individual T-Engine development kits have details.

Q 12 What is the PMC T-Shell?

T-Shell is a collection of middleware running on T-Engine. It includes a GUI manager, for managing the display primitives by which figures and text are drawn in a graphics context, GUI parts (textbox, switches, etc.), menus, windows and the like, a kana-kanji conversion function using VJE, rich character set display functions and multi-language fonts comprising 180,000 characters, TCP/IP manager, and the Microscript visual language ideal for designing touch panels or other user interface screens.

The PMC T-Shell, in addition to its usefulness as GUI middleware in products having an advanced GUI, such as audio-visual systems, office equipment, ticket vending machines, kiosk terminals and the like, offers powerful support as a development platform for embedded applications needing rich character support, such as digital books and dictionaries, or e-government terminals.

A PMC T-Shell development kit is a package software consisting of the PMC T-Shell on a T-Engine development kit. As of December 2005, eleven PMC T-Shell development kits are available for the standard T-Engine specifications (T-Engine/SH7727, T-Engine/SH7751R, T-Engine/SH7760, T-Engine/VR5500, T-Engine/VR5701, T-Engine/TX4956, T-Engine/ARM720-S1C, T-Engine/ARM920-MX1, T-Engine/ARM922-LH7, T-Engine/ARM720-LH7, and T-Engine/ARM926-MB8, T-Engine/ARM926-MX21, T-Engine/PPC-V4FX).

Q 13 What are the terms of use for T-Monitor, T-Kernel, and T-Engine middleware?

The T-Engine Forum (http://www.t-engine.org/) holds the copyright to T-Kernel, the standard real-time OS for T-Engine. The T-Kernel program, including source code, has been made open to the public by the T-Engine Forum. Anyone who agrees to the conditions set forth in the T-License is free to use it, including incorporation in end products.

Note, however, that the T-Kernel offered by the T-Engine Forum lacks MMU support. The addition of MMU support is necessary in order to run the PMC T-Kernel Extension, for realizing file management and virtual memory, or in order to run other middleware such as the PMC T-Shell with its GUI functionality. Personal Media offers the PMC T-Kernel with these and other enhancements.

As for T-Monitor, which is equivalent to the BIOS in a personal computer, this was developed by Personal Media based on specifications of the T-Engine Forum, and is copyrighted by Personal Media. Each T-Engine development kit includes a license to T-Monitor running on that development kit. Purchasers of a development kit may run the included T-Monitor on that product.

Use of T-Monitor in a mass-produced product or on hardware other than the development kit requires a license agreement for this purpose with Personal Media, the T-Monitor developer. The same applies to middleware such as the PMC T-Kernel Extension and PMC T-Shell. Please contact Personal Media regarding specifics, such as the license form and conditions, cost and other details.

Q 14 Are T-Engine technical seminars held?

Seminars concerning T-Engine are presented at various times by Personal Media and other vendors of hardware, middleware, and development tools. Personal Media also offers on-site seminars for customers. These can be matched to the needs and skill levels of the trainees. Contact Personal Media regarding the contents, time, place, cost and other details.

See also the T-Engine Solution Web site page for the latest information.

* T-Engine Training Seminar Practical Introduction to T-Engine for ITRON Engineers Aimed at engineers familiar with ITRON or other real-time OS, this is a technical course providing explanations of T-Engine and T-Kernel, including a description of their role and how they compare with ITRON, process-based programs and MMU use, and other distinctive features of T-Kernel and T-Kernel Extension. The final two hours provide hands-on training in T-Engine development kit use, during which trainees learn about the GNU development environment and the source-level debugger tool gdb.	1. T-Engine Overview 2. T-Kernel Overview and Memory Model 3. PMC T-Kernel Extension Overview 4. Program Development for T-Engine 5. Program Development Training
* T-Engine Expert Seminar This course provides more advanced training that includes development of device drivers and subsystems.	1. Technical Features of T-Engine, T-Kernel, and T-Kernel Extension 2. T-Engine Memory Model 3. T-Engine Device Driver Development 4. T-Engine Subsystem Development

Table 4 T-Engine Seminars: Sample Curriculum

Q 15 Can TCP/IP be used?

A number of TCP/IP products have already been released for use with T-Engine development kits. The PMC T-Shell (* See [Q.12]) also includes TCP/IP functionality. For systems requiring IPv6 support along with IPv4, KASAGO for T-Engine is a TCP/IP protocol stack available from Elmic Wescom, Inc., and sold also by Personal Media. It deserves a serious look by anyone wanting to introduce network functionality in information appliances or other mobile gear, and join the ubiquitous computing age.

In the case of embedded systems such as cell phones, there are various ways to connect to a network (cell phone protocols like CDMA and PHS, as well as wireless LAN schemes and Bluetooth, etc.). Obviously not all connections are going to be made by 10Base-T wired LAN. For this reason, the standard T-Engine and µT-Engine CPU boards do not come with a wired LAN connector or LAN adapter. Customers can choose from the various network connection methods by using the board along with a PCMCIA card (or CF Card in the case of µT-Engine).

Q 16 Can an OS other than T-Kernel (e.g., Linux) be run on the T-Engine board?

A major objective of the T-Engine project is to encourage vendors to make available standard middleware and device drivers. For this reason the project has defined T-Kernel as the standard real-time OS specification. Installing a different OS on a T-Engine board would be contrary to that objective and is not recommended, since the standards for middleware and device drivers would no longer apply.

Technically, of course, it is possible to install another OS on a T-Engine board. T-Monitor can be made to boot a program or OS other than T-Kernel, as long as the specified boot format is followed when installing the program in flash memory or on a PC Card (ATA card).

Another approach is to implement Linux as a huge middleware program running on T-Engine and T-Kernel. This approach is adopted in T-Linux. Since the OS kernel is replaced by T-Kernel, this approach offers better real-time performance than ordinary Linux systems, and T-Linux can easily be used along with T-Engine device drivers and middleware. Details of T-Linux can be obtained by contacting the T-Engine Forum.

Q 17 Do I need to take eTRON into consideration when developing middleware, etc., since T-Engine has, as one of its features, an interface with eTRON?

If the software you are trying to develop does not use eTRON functions, you do not have to take eTRON into account. You can develop your software independent of eTRON. On the other hand, you can develop software to manage the accounting or copyright protection with making the most of eTRON. Please contact the T-Engine secretariat for information on development with eTRON.

Q 18 Is there any restrictions regarding the business structure of middleware for T-Engine?

Not in particular. Since the T-Engine project standardizes only the specifications and interfaces, each embedded product manufacturer has complete latitude in its handling of its own products or business structure. By the same token, there is no restriction on the structure of providing middleware that run on T-Engine. The makers are also free in terms of provision of source code or royalty collection. The T-Engine project is expecting various types of products to be developed, including the ones that are free and open(for modifications or education purposes), and those that require some payment but of high quality.

*Reference Books and Related Information

- T-Kernel Kumikomi Programing Kyokasho (A Practical Guide to T-Kernel Embedded Programing), Ken Sakamura (Ed.), 4200yen.
- T-Kernel Standard Handbook, Ken Sakamura (Ed.), 3800yen.
  * English documents are included as PDF files on CD-ROM.
- T-Engine, T-Engine2, T-Engine3 (TRONWARE extra edition), 1600yen (each).
- TRONWARE Vol.1-109 (TRON bimonthly magazine, scheduled to be continued), 1200yen (ordinary issue).
- µITRON 4.0 Standard Guidebook, Ken Sakamura (Ed.), 3200yen.

* The above books are published by Personal Media Corporation in Japanese. Tax is not included.

- T-Engine Forum	http://www.t-engine.org/
- TRON Project	http://www.tron.org/
- Personal Media Corporation (T-Engine Solution Website)	http://www.t-engine4u.com/

Personal Media Corporation

Koizumi Bldg. 1-29-1 Nishi-Gotanda, Shinagawa-ku, Tokyo 141-0031 Japan
E-mail: te-sales@personal-media.co.jp
http://www.personal-media.co.jp/welcome-e.html
Tel: +81-3-5759-8305
Fax: +81-3-5759-8306

- For details and price, please contact us.
- TRON is an abbreviation of "The Real-time Operating System Nucleus".
- eTRON is an abbreviation of "entity and economy TRON".
- TRON, eTRON, T-Engine, µT-Engine, T-Monitor, and T-Kernel are specified terms for computers, and are not product names.
- All product names are trademarks or registered trademarks of their respective owners.
- Due to continued product upgrade or enhancement, the information in this document is subject to change without notice.
- T-Engine is an open, royalty-free specification, and not a commercial product.