

An Open Development Platform for Embedded Systems

# T-Engine Development Kit



**The T-Engine Development Kit is an open platform for embedded system development, offered as a total package complete with hardware, software, and development environment.**

---

#### Embedded development issues...

**Efficiency is a must for embedded software development.**

- The addition of networking capability to embedded systems has greatly increased their features.
- The products themselves are becoming more sophisticated.
- Products must be given competitive features without raising costs.

**But the reality is different.**

- Development needs to be completed in a short time.
- Development cycles follow on each other at a dizzying pace.
- Starting development from scratch is not an option.
- Standards are slow to emerge, making reuse difficult.
- Middleware and driver availability is spotty.

---

#### The T-Engine Development Kit solution...

- The standardized real-time kernel, boasting high performance and compact code size, as well as the hardware, software, and development environment can be used with a wide range of CPU architectures.
- A wealth of software components are readily available at low cost, reducing development cost and time.
- The standard, open architecture simplifies technical training. Knowledge of T-Engine and T-Kernel can be put to good use in future development.

## A standardized architecture

### Hardware specifications:

Not only the basic specifications but matters such as physical connector position and board size are standardized.

### Software specifications:

Comes with a standard real-time kernel (T-Kernel) and monitor (T-Monitor, similar to the BIOS in a PC).

### Development environment:

Comes with a GNU-based open development environment. Capable of generating the T-Engine standard object format. This standardization of hardware, software, and development environment makes it easier to provide middleware and to recycle software assets. And since a common training program can be applied to all technical staff involved with T-Engine, expertise once acquired is good for use with other T-Engine products.

## A wide choice of CPU architectures

Since the hardware, OS and other software, and development environment specifications are all standardized, a changeover to a different CPU can be accommodated readily, with little more than software recompiling.

Vendors are therefore free to choose the optimum CPU for each application, based on factors such as performance and power consumption.

## A rich assortment of middleware and development tools

The standard architecture means middleware and development tools are plentiful and available at low cost.

### Advantages for software vendors:

A growing market is created for middleware of various kinds. Supporting T-Engine products with different CPUs is simply a matter of recompiling.

### Advantages for peripherals vendors:

A variety of peripheral devices can be connected by commonly used interfaces such as USB, PC Card (standard T-Engine), CF Card, and MMC ( $\mu$ T-Engine). Support for T-Engine products with different CPUs requires little more than recompiling the common device driver software.

### Advantages for embedded products vendors:

Embedded control software for feature-rich products can be developed in a short time by combining middleware from the wide assortment of packages available at low cost.

### Advantages for development tools vendors:

T-Engine represents a promising market for GUI-based visual development tools, debuggers, and integrated development environments. Vendors can also build and market development tools with support for additional hosts of their own choosing.

## A wealth of peripheral devices

A standard T-Engine board comes with an array of standard interfaces including serial, PC Card, USB, touch panel, LCD, sound, and expansion bus interfaces. This means, of course, that a wide variety of peripheral devices can be connected. Moreover, public disclosure of the expansion board specifications makes it possible to expand the system in various ways, by means of a CPU board or other expansion boards.



## A diverse series lineup

Standard T-Engine is available for products requiring a GUI, whereas compact control applications that do not need a GUI can be created using  $\mu$ T-Engine, giving vendors a choice of series based on the application field.

## A wide range of uses

### Carried around

The physically compact and stable hardware configuration means software development and debugging can be performed with confidence. The development environment itself can be carried around, and used for debugging or data gathering at different locations.



T-Engine Development Bench (portable acrylic cover)

### As a development evaluation board

The T-Engine board can be used as a reference platform for evaluation when developing embedded system control software. The wide range of application fields includes cell phones, audio-visual products, and digital appliances.

### For middleware development

As an executable target board, it can be used for developing, testing and debugging middleware and application software running on T-Engine.

### As a compact board computer for test and training use

The product includes not only software specifications but circuit diagrams and other detailed hardware information, making it an ideal board computer for experimental and training use.

## Target-side Software

### PMC T-Kernel

A real-time kernel implementing the T-Kernel specifications and configured as follows.

#### T-Kernel/Operating System (T-Kernel/OS)

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

#### T-Kernel/System Manager (T-Kernel/SM)

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

#### T-Kernel/Debugger Support (T-Kernel/DS)

- Kernel internal state referencing
- Trace

Ordinarily "T-Kernel" refers to all of T-Kernel/OS, T-Kernel/SM, and T-Kernel/DS; but it is also sometimes used in the narrow sense to mean T-Kernel/OS only.

### PMC T-Kernel Extension<sup>(\*)</sup>

Sometimes called basic development middleware, this is a set of OS extensions for the Development Kit built using T-Kernel/OS subsystem functions.

- Process/task management
- Message management
- Process/task synchronization and communication management

- Global name management
- Memory management
- File management
- Event management
- Device management
- Time management
- System management

These extensions realize a virtual memory OS, providing file and process functions as part of the development environment.

### PMC T-Monitor

A monitor implementing the T-Monitor specification. T-Monitor is burned in ROM as the basic T-Engine monitor providing the following functions.

#### 1. System functions

- Hardware initialization
- System boot
- Exception/interrupt/trap handling functions

#### 2. Debugging functions

- Memory operations
- Register operations
- I/O operations
- Disassembly
- Program and data loading
- Program execution
- Breakpoint operations
- Trace execution
- Disk read/write/boot operations

#### 3. Program support functions

- Monitor service function provision

### Device drivers

The following device drivers are included for Development Kit use, based on the T-Kernel/SM device management function specifications.

1. Clock (RTC) driver
2. Console (serial) driver
3. System disk drivers (ATA/CF card, USB interface disk)<sup>(\*)</sup>
4. KB/PD drivers (keyboard, touch panel, USB keyboard/mouse)<sup>(\*)</sup>
5. Screen (built-in LCD) driver<sup>(\*)</sup>
6. PC Card manager (bus driver)
7. USB manager (bus driver)<sup>(\*)</sup>

### Development tools

#### IMS (Initial Monitor System)

A program started as the T-Kernel initial task.

- Command-based operations for referencing and manipulating T-Kernel states
- System program (subsystem) loading and unloading
- System program (process) execution
- Command file execution
- Auto-execution of STARTUP.CMD, the system initialization command file

#### CLI (Command Line Interpreter)<sup>(\*)</sup>

A program runs as a system process.

- Command-based operations mainly on files
- System program (subsystem) loading and unloading
- Application program execution
- Command file execution

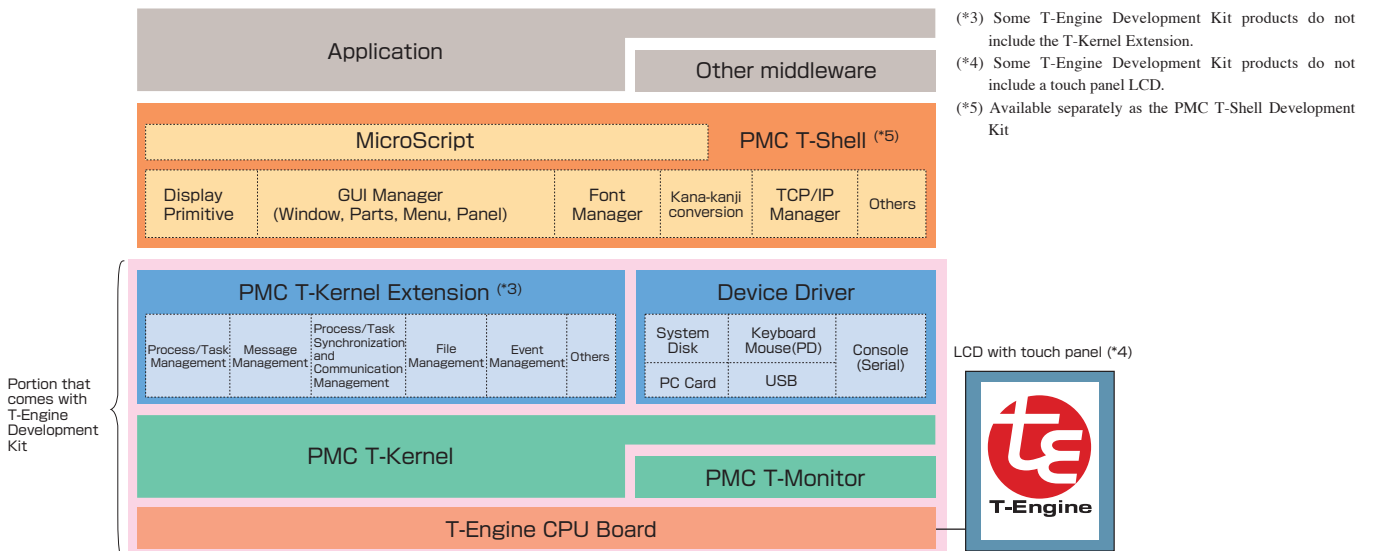
#### Utilities

Various tools, test programs, etc.

- The detailed software configuration differs with the T-Engine product. See the literature for individual products.
- See the individual product literature for the particular hardware configuration.

(\*1)Not included with some  $\mu$ T-Engine models.

(\*2)Not included with  $\mu$ T-Engine.



## Host-side Software (Cross-development Environment)

1. A GNU-based cross-development environment running on Linux and Windows (Cygwin) is provided as host-side software.
2. This is a cross-development environment built using GNU tools (gcc, binutils, gdb).  
It has been confirmed to run on RedHat Linux 7.1 / 7.3 / 8.0 / 9.0 and Cygwin on Windows XP Professional
3. It comes with an sample assortment of sample source code. Device drivers, middleware, and application software for

T-Engine can be built easily while referencing the sample source code.

4. It consists of the following development environment packages and source packages.

### GNU development environment packages

- Common parts for Linux and Cygwin each.
- Target CPU-specific parts for Linux and Cygwin each.
- T-Kernel resources (including drivers and sample source code)

## Eclipse for PMC T-Kernel

"Eclipse for PMC T-Kernel" is an GUI-based integrated development environment. By operating GUI, you can proceed a series of development such as coding, compiling and debugging.

"Eclipse for PMC T-Kernel" accepts a wide variety of CPUs, including ARM, MIPS, SH, x86 and PowerPC.

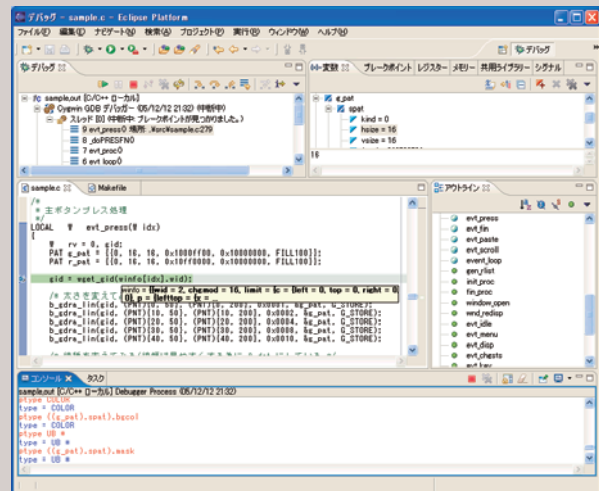
You can use the same development environment, even if you use another CPU.

In other words, you don't have to pay attention to the difference of CPUs.

Thereby, you can easily change the CPU for your future project.

The registered user of our T-Engine Development Kit and T-Engine Appliance can use "Eclipse for PMC T-Kernel" for free.

\*The Eclipse-based Development Environment is only in Japanese, including plug-in itself and manual. English version is not yet available.



## Middleware for T-Engine

Personal Media Corporation (PMC) has many years of experience as a developer of full-featured operating systems and applications that are event-driven, offer real-time performance, support multi-tasking, and are GUI-based. Drawing on this experience, we are now providing middleware

and application software for T-Engine. Developers of T-Engine systems are invited to take advantage of this software. In addition, PMC stands ready to develop software to individual customer specifications.

## T-Engine Solutions

Personal Media Corporation (PMC) offers total support for developing products based on T-Engine, including porting and tuning T-Monitor (BIOS), T-Kernel, and device drivers as

needed for custom hardware. Contact PMC about T-Engine porting and tuning, application system building, and other related matters.

## T-Engine Seminars

Personal Media Corporation (PMC) hosts T-Engine seminars periodically as part of our effort to promote understanding and wide acceptance of T-Engine. The "T-Engine Introductory Seminars" focus on the distinctive features of T-Kernel and T-Kernel Extension, giving practical explanations along with hands-on training. The "T-Engine Expert Seminars" take up

more advanced topics, including device driver and subsystem design. Sessions are held monthly, with the schedule announced on the T-Engine Solutions Website. Customers interested in special customized seminars are invited to contact PMC about details of time, place, and seminar curriculum.

Software development for electronic devices and embedded systems has grown increasingly difficult as these products become more sophisticated and shorter development time is demanded. Meanwhile, standards for embedded system software development environments have been slow to emerge, impeding efforts to provide a sufficient supply of middleware, device drivers and other software components.

The T-Engine project was started to address these issues by standardizing the development platform for embedded systems, extending to hardware and development tools, thereby encouraging greater availability of software components and making them more portable.

To run the project, the T-Engine Forum (<http://www.t-engine.org/>) was formed, made up of Japan's leading semiconductor manufacturers, software vendors (including those with parent companies overseas), embedded systems vendors, home electronics manufacturers and many

other companies. This organization is engaged in R&D on the T-Engine architecture as well as carrying on standardization activities.

T-Engine is part of the TRON Project. Real-time OS standardization for embedded systems was originally carried out in the ITRON (Industrial TRON) subproject, resulting in the appearance of OSEs that are at the heart of cell phones and numerous other embedded systems products, and that have been a major foundation of the information technology industry in Japan.

OS standardization alone, however, can only go so far toward facilitating software component availability and portability. The T-Engine project, by devising tighter standards that apply not only to OS service call specifications but also to hardware specifications, software interfaces, object formats and other matters, aims to achieve more effective utilization of software assets, enabling them to be used as a common platform.

### T-Engine Hardware Specification

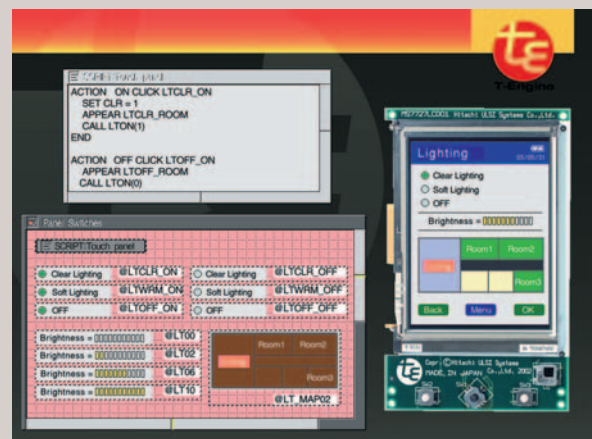
	Standard T-Engine	μT-Engine
CPU	32 bit	
MMU	required	optional
RAM	16MB or higher	4MB or higher
Flash memory	4MB or higher	
Serial I/O	38,400bps or higher	
Calendar clock	Yes	
Sound CODEC	Yes(IN:1ch, OUT:2ch)	No
eTRON Chip I/F	Yes	
LCD Panel I/F	Yes	No
Touch panel I/F	Yes	No
Extension bus I/F	Yes	
Other I/Fs	PCMCIA Slot Type-II x 1 USB Host Type A connector x 1	CF Card Slot x 1 MMC Card Slot x 1
Board size	75mm x 120mm	60mm x 85mm

### 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/en/index.html>

## PMC T-Shell Development Kit

The PMC T-Shell Development Kit is middleware for T-Engine that enables provision of a GUI system with rich character support. It includes screen drawing functions, GUI parts, a windowing system and other GUI functions, as well as kana-kanji transform, TrueType fonts with more than 180,000 characters, and a TCP/IP manager. It can be used to develop GUI-based applications quickly and easily. Moreover, it comes with the visual language MicroScript, allowing programs developed for a PC using MicroScript to be run on T-Engine with little or no modification.



Standard T-Engine Board



SH3-DSP (RENESAS)

T-Engine/SH7727 Development Kit



SH-4 (RENESAS)

T-Engine/SH7760 Development Kit



MIPS

T-Engine/Vr5701 Development Kit



MIPS (TOSHIBA)

T-Engine/TX4956 Development Kit



PowerPC405 (Xilinx)

T-Engine/PPC-V4FX Development Kit



ARM9 (Freescale i.MX)

T-Engine/ARM920-MX1 Development Kit



ARM9 (Freescale i.MX)

T-Engine/ARM926-MX1 Development Kit



ARM9 (FUJITSU)

T-Engine/ARM926-MB8 Development Kit



ARM9 (SHARP)

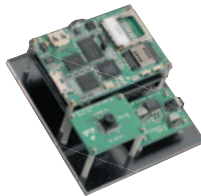
T-Engine/ARM922-LH7 Development Kit



ARM7 (SHARP)

T-Engine/ARM720-LH7 Development Kit

μT-Engine Board



M32R (RENESAS)

μT-Engine/M32104 Development Kit



MIPS (NEC)

μT-Engine/Vr4131 Development Kit



V850 (NEC)

μT-Engine/V850E-MA3 Development Kit



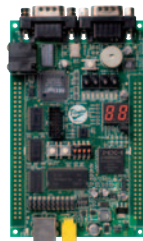
ARM7 (SHARP)

μT-Engine/ARM7-LH79532 Development Kit

T-Engine Appliance



Teacube/Vr5701 Evaluation Kit



Teaboard2/ARM920-MX1



μTeaboard/ARM920-MX1



T-Kernel/x86 Development Kit

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/  
 Tel: +81-3-5759-8305  
 Fax: +81-3-5759-8306

- Teacube is a registered trademark of Personal Media Corp.
- 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.