Getting Started with T-Kernel 2/x86

Version 2.0.1

Personal Media Corporation

Contents

History of Changes

Version 2.0.1

- Modify “1. Installing Target” by updating target-side environment.
- Modify “2. Installing Development Environment” by updating development environment.

Version 2.0.0

- First edition for T-Kernel 2/x86 Evaluation Kit.

Introduction

This document provides a tutorial on starting out with T-Kernel 2/x86 Evaluation Kit, for those using this product for the first time.

1 Installing Target

1.1 Real and Virtual Environments

In this section we'll install the target-side environment (OS, drivers and middleware).

In general on developing embedded systems, the development environment is apart from the execution target. This style is called "cross development". A typical "real development environment" is shown in Figure 1.1, which supports T-Kernel 2/x86 Evaluation Kit. In addition, T-Kernel 2/x86 also supports another type of development environment, namely "virtual environment" as shown in Figure 1.2, where the target-side runs on VMware.

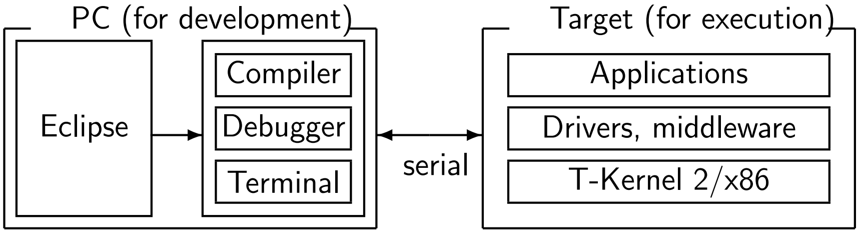


Figure 1.1 Real Environment

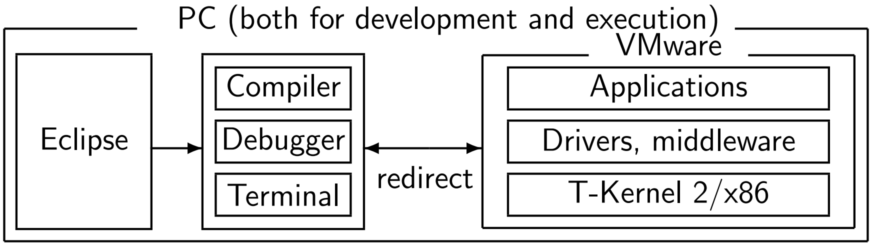


Figure 1.2 Virtual Environment

The end product of embedded systems should be on the real target machine in almost cases. Virtual environment is, however, also useful in software development phases, because:

- Only one PC is required in software development phases.
- One can start software development even when the target machine hardware is not yet developed.

If you prefer virtual environment, follow sections “1.2 Installing VMware Player”, “1.3 Installing VMwareGateway” and “1.4 Installing T-Kernel/x86”.

On the other hand, if you prefer real environment, VMware Player and VMwareGateway are not required. Skip these sections and go to “1.4 Installing T-Kernel/x86”.

Before installation please also refer to *“T-Kernel 2/x86 Evaluation Kit Manual”* contained in the T-Kernel 2/x86 Evaluation Kit CD-ROM, and look at “1.2 Hardware Requirements” to see that your machine conforms to the hardware requirements.

1.2 Installing VMware Player

VMware Player is required to run T-Kernel 2/x86 on virtual environment.

(1) Start Installer

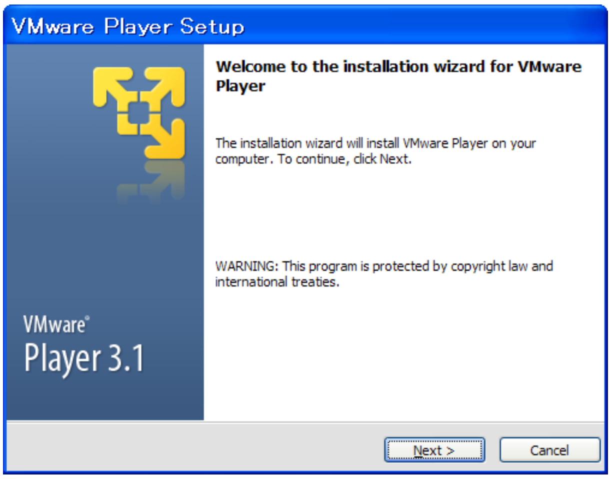
Start “`VMware-player-3.1.6-744570.exe`” on the common software folder “`common\soft`” in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

† The version number in the file name may be different.

† If you want to install other version of VMware Player instead of that in the T-Kernel 2/x86 Evaluation Kit CD-ROM, please download suitable version for your host PC from the following website:

`http://www.vmware.com/`

After a moment the Next button becomes enabled. Then click on this button. (Figure 1.3)



VMware Player Setup

Welcome to the installation wizard for VMware Player

The installation wizard will install VMware Player on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure 1.3 Start dialog of VMware Player Installer

(2) Select Install Folder

Select install folder. No needs to change the default folder. Click on Next.

(3) Make Shortcut

No needs to change the default.
Just click on Next.

(4) Start Installation

Click on Continue to start installation, and wait for completion.

(5) Installation Completed

After completion, click on Restart Now or Restart Later and restart Windows.

1.3 Installing VMwareGateway

In case of real environment, RS-232C (serial connection) is used to connect console between the host PC and the target machine. In case of virtual environment, instead, socket connection is used. To use this socket connection, VMwareGateway is required. It is installed as a “service” on Windows.

For details please also refer to *“VMware Support Tool Install Manual”* contained in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

(1) Make Install Folder

Create “`utilities`” folder under “`C:\Program Files\VMware`”.

(2) Copy Executable File

Copy “`vmwaregateway.exe`” from the common software folder “`common\soft`” to “`C:\Program Files\VMware\utilities`”.

(3) Register VMwareGateway Service

Run “Command Prompt” of Windows.

† In cases of Windows 8, 7 and Vista, run Command Prompt as an administrator by right-click menu.

And execute “`C:\Program Files\VMware\utilities\vmwaregateway.exe`” with “`/r`” option. Note that the path name must be quoted by `"···"`, because the path includes spaces. (Figure 1.4)

```
C:\Documents and Settings\admin> "C:\Program Files\VMware\
utilities\vmwaregateway.exe" /r ↩
Create VMWare Gateway Service for executable C:\Program Fi
les\VMware\utilities\vmwaregateway.exe
CreateService SUCCESS
C:\Documents and Settings\admin> exit ↩
```

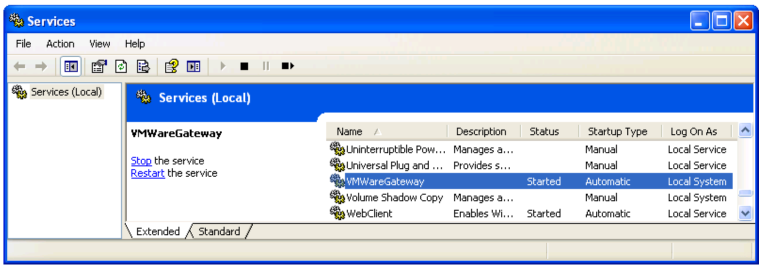
Figure 1.4 Register VMwareGataway Service

(4) Start VMwareGateway Service

Open "Services".

† Windows 8 and 7 : "Control Panel" → "System and Security" → "Administrative Tools" → "Services"
Windows Vista : "Control Panel" → "System and Maintenance" → "Administrative Tools" → "Services"
Windows XP : "Control Panel" → "Performance and Maintenance" → "Administrative Tools" → "Services"

Select "VMwareGateway" in the list of services, and select "Start" of right-click menu. (Figure 1.5)



VMWareGateway

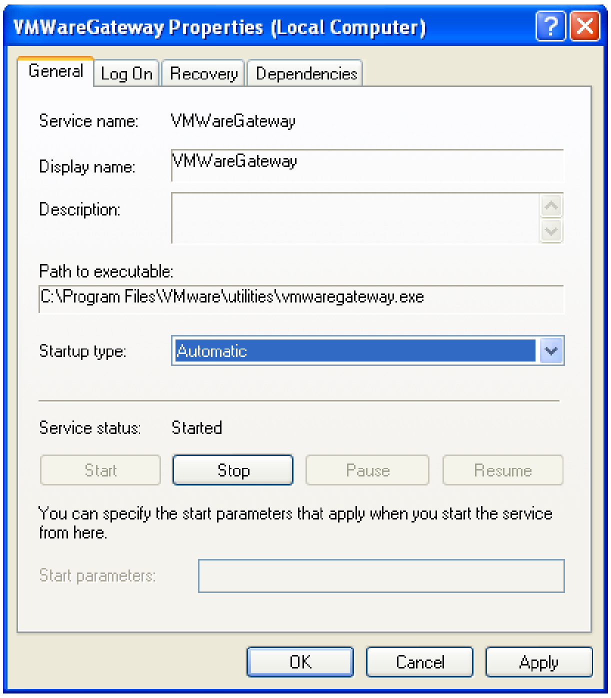
Stop the service
Restart the service

Name	Description	Status	Startup Type	Log On As
Uninterruptible Pow...	Manages a...		Manual	Local Service
Universal Plug and ...	Provides s...		Manual	Local Service
VMWareGateway		Started	Automatic	Local System
Volume Shadow Copy	Manages a...		Manual	Local System
WebClient	Enables Wi...	Started	Automatic	Local Service

Figure 1.5 Start VMwareGateway Service

(5) Automatic Start VMwareGateway

Select "VMwareGateway", and open its property by right-click menu. Set "Startup type" to "Automatic". (Figure 1.6)



VMWareGateway Properties (Local Computer)

Service name: VMWareGateway

Display name: VMWareGateway

Description:

Path to executable:
C:\Program Files\VMware\utilities\vmwaregateway.exe

Startup type: Automatic

Service status: Started

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Figure 1.6 Automatic Start VMwareGateway

This completes the installation of VMwareGateway.

1.4 Installing T-Kernel 2/x86

Install T-Kernel 2/x86 on virtual environment (VMware) or a real target machine. For details please also refer to *"T-Kernel 2/x86 Evaluation Kit Manual"* at "2. Target-side Software Installation".

For this installation a terminal software is required, such as "HyperTerminal" and "Tera Term Pro".

† In cases of Windows 8, 7 and Vista, where HyperTerminal is not included, please download "Tera Term Pro (version 2.3)" from the following website:

`http://hp.vector.co.jp/authors/VA002416/teraterm.html`

(1) Copy Virtual Machine (required only in case of virtual environment)

- **In case of virtual environment (VMware) :**

Copy the virtual machine folder “`tkx86_vm`” from the “`jp`” folder of T-Kernel 2/x86 Evaluation Kit CD-ROM to the “`My Virtual Machines`” folder in “`My Documents`”.

† If there is no “`My Virtual Machines`” folder in “`My Documents`”, please create it.

- **In case of real environment :**

  No needs to copy.

(2) Start Terminal Software

- **In case of Tera Term Pro :**

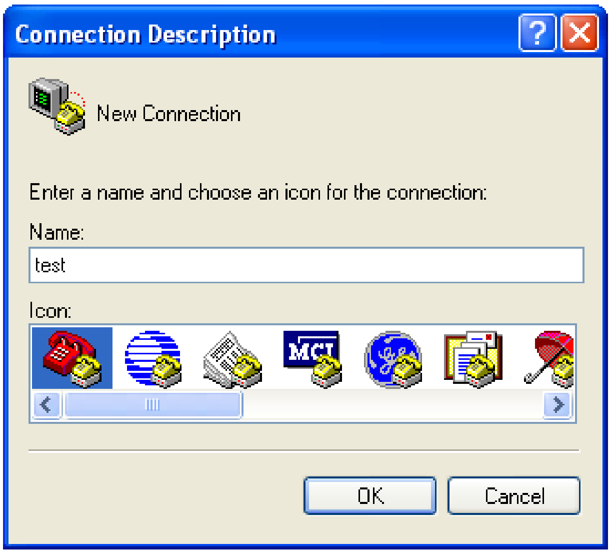
  Start Tera Term Pro.

- **In case of HyperTerminal :**

  Start “HyperTerminal” (by clicking on “Start button” → “All Programs” → “Accessories” → “Communications” → “HyperTerminal”).

  † You may be asked about “Default Telnet Program”, but it is not significant. Phone number is also not significant, because no telephone call is used actually.

  Enter any name and select any icon, and click on OK. (Figure 1.7)



Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name:

test

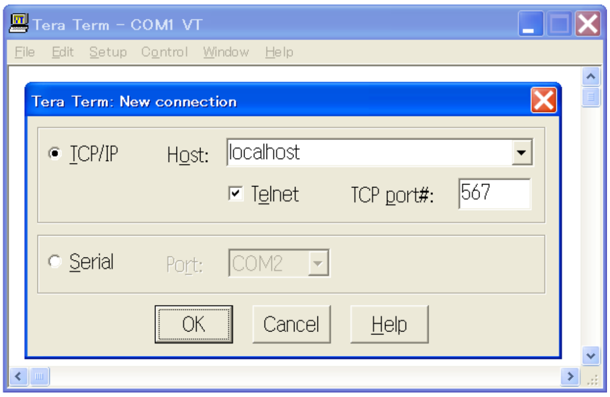
Icon:

Figure 1.7 Start HyperTerminal

(3) Select Connection Type

- **In case of virtual environment (VMware) :**
  Use telnet connection by VMwareGateway. Connect to the host "`localhost`" and port number `567`.

  - **In case of Tera Term Pro :**
    Select "File" → "New connection". On the new connection dialog, check "TCP/IP", enter "`localhost`" as host, check "Telnet", and enter "`567`" as TCP port#. (Figure 1.8)



Tera Term: New connection

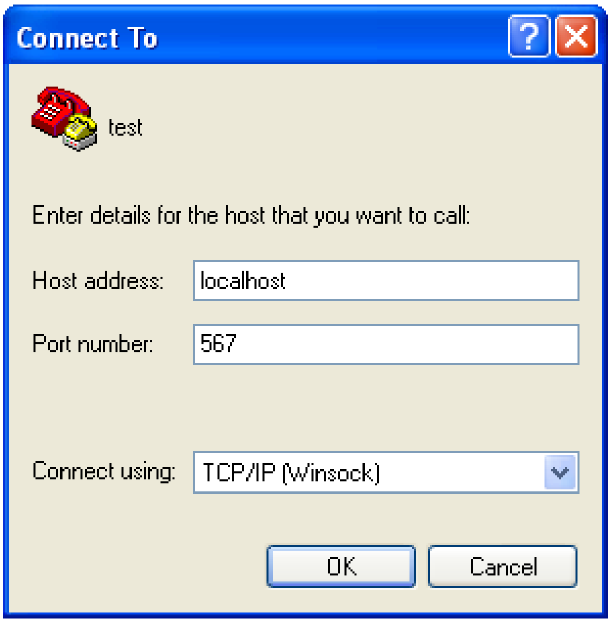
● TCP/IP Host: localhost

☑ Telnet TCP port#: 567

○ Serial Port: COM2

Figure 1.8 Telnet Connection by Tera Term Pro

  - **In case of HyperTerminal :**
    Select "TCP/IP (Winsock)TCP/IP", enter "`localhost`" as host address, and enter "`567`" as port number. Click on $\boxed{\text{OK}}$. (Figure 1.9)



Connect To

test

Enter details for the host that you want to call:

Host address: localhost

Port number: 567

Connect using: TCP/IP (Winsock)

Figure 1.9 Telnet Connection by HyperTerminal

- **In case of real environment :**

  Connect your host PC and the target machine by serial connection (RS-232C). Select serial port of the host PC (COM1 etc.), and set the communication parameters as follows:

Speed	115200 bps
Data bits	8 bits
Parity	none
Stop bits	1 bit
Flow control	hardware

(4) Boot target from CD-ROM

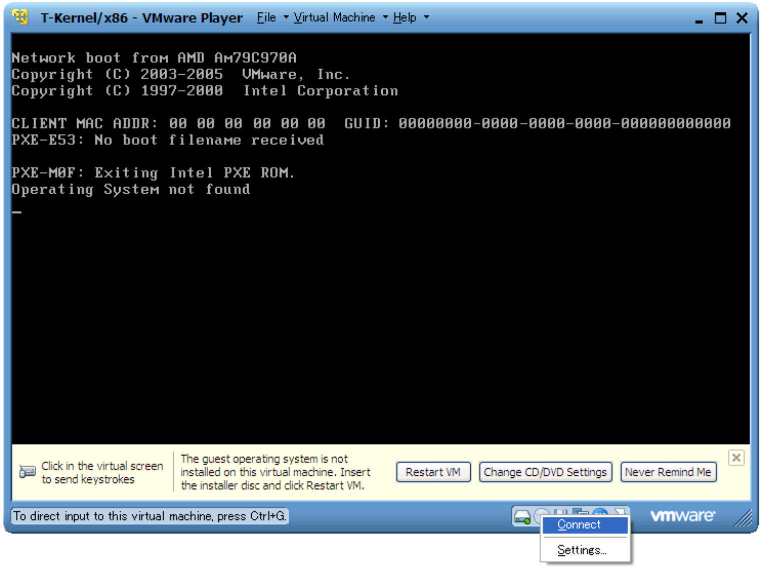
- **In case of virtual environment (VMware) :**

  Insert T-Kernel 2/x86 Evaluation Kit CD-ROM to the CD-ROM drive of your PC.

  Run the virtual machine “`T-Kernel_x86.vmx`” in
  ”`My Documents\My Virtual Machines\tkx86_vm`”.

  VMware Player’s window is opened, but then appears a message “Operating System not found”. It is because the CD-ROM drive is not yet connected to the virtual machine. Then click on “CD/DVD(IDE)” icon on the bottom side of the

VMware Player's window (or VMWare Player's toolbar → "Virtual Machine" → "Removable Devices" → "CD/DVD(IDE)") and select "connect". (Figure 1.10)



```
Network boot from AMD Am79C970A
Copyright (C) 2003-2005  VMware, Inc.
Copyright (C) 1997-2000  Intel Corporation

CLIENT MAC ADDR: 00 00 00 00 00 00  GUID: 00000000-0000-0000-0000-000000000000
PXE-E53: No boot filename received

PXE-M0F: Exiting Intel PXE ROM.
Operating System not found
_
```

Click in the virtual screen to send keystrokes

The guest operating system is not installed on this virtual machine. Insert the installer disc and click Restart VM.

To direct input to this virtual machine, press Ctrl+G.

Figure 1.10 Connect CD-ROM drive to VMware Player

† Even if you use USB-connected CD/DVD drive, you have to select "CD/DVD(IDE)" instead of USB device.

Click on VMware Player's window to erase the mouse cursor arrow.

Push Ctrl + Alt + Insert to reboot the virtual machine from the CD-ROM.

- **In case of real environment :**

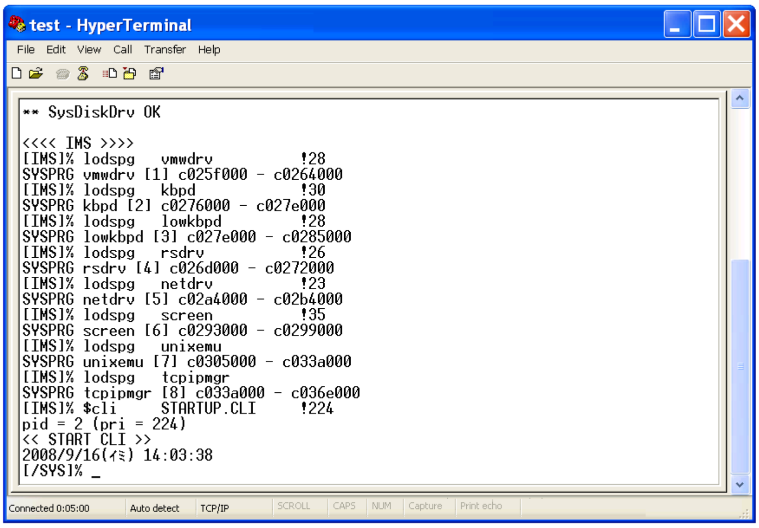
  Insert T-Kernel 2/x86 Evaluation Kit CD-ROM to the CD-ROM drive of the target machine, and boot from the CD-ROM.

  † Check BIOS settings to enable CD-ROM booting.

(5) Confirming Prompt

The target machine's display (VMware Player's window in case of virtual environment, or video display in case of real target machine) shows a T-Kernel logo graphic and then a console screen.

On the other hand, the terminal software of the host PC shows the CLI prompt "`[/SYS]%`" by entering ↩(Enter) key. (Figure 1.11)



```
** SysDiskDrv OK

<<<< IMS >>>>
[IMS]% lodspg   vmwdrv          !28
SYSPRG vmwdrv [1] c025f000 - c0264000
[IMS]% lodspg   kbpd            !30
SYSPRG kbpd [2] c0276000 - c027e000
[IMS]% lodspg   lowkbpd         !28
SYSPRG lowkbpd [3] c027e000 - c0285000
[IMS]% lodspg   rsdrv           !26
SYSPRG rsdrv [4] c026d000 - c0272000
[IMS]% lodspg   netdrv          !23
SYSPRG netdrv [5] c02a4000 - c02b4000
[IMS]% lodspg   screen          !35
SYSPRG screen [6] c0293000 - c0299000
[IMS]% lodspg   unixemu
SYSPRG unixemu [7] c0305000 - c033a000
[IMS]% lodspg   tcpipmgr
SYSPRG tcpipmgr [8] c033a000 - c036e000
[IMS]% $cli     STARTUP.CLI     !224
pid = 2 (pri = 224)
<< START CLI >>
2008/9/16(ｲﾐ) 14:03:38
[/SYS]% _
```

Figure 1.11 Prompt showed on Terminal Software

† The console screen on the target machine's display also shows the CLI prompt, and you can use it. But it is recommended to do the installation process below on the terminal software of the host PC, instead of the target machine's display.

(6) Making a Partition

Run `hdpart` command from the terminal software, create partition and set boot partition on the disk. The device name is "`hda`", which means the whole disk.

```
[/SYS]% hdpart -m1 hda ↩                                  — Run hdpart command
hda    [C:1017  H:255  S:63  B:16357785 (7987 MB)]
No System    Boot StartCHS      EndCHS    SecNo  SecCnt     Size
1  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
2  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
3  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
4  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
** Create/Delete/Boot/Edit/Quit ? c ↩                      — Create partition
Create PartNo (1-4) ?  1 ↩                                  — 1st partition
Size [GB/MB/KB,All] (<7987MB) ? a ↩                                    — All
No System    Boot StartCHS      EndCHS    SecNo  SecCnt     Size
1  13 BTRON  00    0:   1: 1  1018:  57:24    63 16357722 7987 MB
2  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
3  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
4  00 ------ 00    0:   0: 0     0:   0: 0     0        0     0 KB
** Create/Delete/Boot/Edit/Update/Quit ? b ↩                          — Boot
Boot PartNo (1-4,Clear) ? 1 ↩                                — 1st partition
No System   Boot  StartCHS      EndCHS    SecNo  SecCnt     Size
```

```
1  13 BTRON  80   0:   1: 1 1018: 57:24    63 16357722 7987 MB
2  00 ------ 00   0:   0: 0    0:  0: 0     0        0    0 KB
3  00 ------ 00   0:   0: 0    0:  0: 0     0        0    0 KB
4  00 ------ 00   0:   0: 0    0:  0: 0     0        0    0 KB
** Create/Delete/Boot/Edit/Update/Quit ? u ↩                 — Update
** hda: Updated Master Boot Block
[/SYS]%
```

† If you want to install T-Kernel 2/x86 Evaluation Kit to an existing partition, delete it by "`d`" (=delete) and create it again by "`c`" (=create).

† If you want to version-up from the previous product (T-Kernel/x86 Evaluation Kit), please also delete the partition of the previous product, create a new partition, and then install the new product.

(7) Format

Format the created partition by `formatsys` command from the terminal software. Device name is "`hda0`" for 1st partition, "`hda1`" for 2nd partition, and so on. And "`-b`" option (write boot record) is required. Also "`-x`" option is usually required.

```
[/SYS]% formatsys -b -x hda0 SYSTEM ↩          — Run formatsys command
Format pcb0 [STD] SYSTEM
Logical Formatting...
Writing BootCode...
Disk Format Success.
.
ATYPE ATR   NREC  NREF SIZE BLK MTIME           NAME
0003 ----- 1 0    1  35812  10 121103 16:23:39 SBOOT
0003 ----- 1 0    1 242847  61 121103 16:23:40 KERNEL.SYS
0003 ----- 1 0    1 109984  28 121103 16:23:40 KDRIVER
0003 ----- 1 0    1   4799   3 121103 16:23:40 SYSCONF
0003 ----- 1 0    1   1479   2 121103 16:23:40 DEVCONF
[/SYS]%
```

† You must use `formatsys` command here (*NOT* `format` command).

† Date and time of each files are current time at installation.

(8) Attach Target Disk

Attach the target disk, which is formatted in the last step.

```
[/SYS]% att hda0 HD ↩                                    — Attach disk
hda0 -> HD
```

```
[/SYS]% cd /HD ↩                                — Change current directory
[/HD]% ls -l ↩                                              — List of files
ATYPE ATR  NREC  NREF SIZE BLK MTIME           NAME
0003 ----- 1 0    1  35812  10 121103 16:23:39 SBOOT
0003 ----- 1 0    1 242847  61 121103 16:23:40 KERNEL.SYS
0003 ----- 1 0    1 109984  28 121103 16:23:40 KDRIVER
0003 ----- 1 0    1   4799   3 121103 16:23:40 SYSCONF
0003 ----- 1 0    1   1479   2 121103 16:23:40 DEVCONF
```

[**Trouble Shooting**] If there are no files on `/HD`, it means that your installation process was not correct. Please change directory by "`cd /SYS`", detach the disk by "`det hda0`", and do the installation process again.

(9) Attach CD-ROM

Use `df` command to check the device name of CD-ROM, which contains the install data. For example, if `/SYS` is `hdc2`, the install data is contained in its #0 partition, namely `hdc0`. Then attach it.

```
[/HD]% df ↩                                                 — Disk status
PATH DEV    TOTAL    FREE USED UNIT MAXFILE NAME
/SYS hdc2    3100      22  99% 1024     256 PM:TK2-x86
/HD  hda0 8178860 8176304   0% 4096   65535 SYSTEM
[/HD]% att -c hdc0 CD ↩                                  — Attach CD-ROM
hdc0 -> CD (ReadOnly)
```

(10) Installing the Basic System

Expand and run the basic system installer.

```
[/HD]% expf -U /CD/EN/SOFT/INSTALL.BZ ↩                  — Expand archive
[/HD]% install /HD ↩                                        — Run installer
Basic system installation completed.
[/HD]% rm -r install ↩                                   — Remove installer
```

(11) Installing PMC T-Shell

Expand and run the T-Shell installer.

```
[/HD]% expf -U /CD/EN/SOFT/TSHELL.BZ ↩                   — Expand archive
[/HD]% install_tshell /HD ↩                                 — Run installer
Install completed, please reboot the system.
[/HD]% rm -r install_tshell ↩                            — Remove installer
```

Do not reboot the system now before installing the English patch in the next step.

(12) Installing the English Patch

Expand and run the English patch. This patch translates T-Shell GUI applications to English.

```
[/HD]% expf -U /CD/EN/SOFT/ENGLISH.BZ ↩                  — Expand archive
[/HD]% english /HD ↩                                        — Run installer
..  Installing English Patch on /HD
..  Install completed
[/HD]% rm -r english ↩                                   — Remove installer
```

(13) Reboot T-Kernel 2/x86

Use `exit` command to shutdown the target.

```
[/HD]% exit ↩                                                    — Exit CLI
<< EXIT cli >>
[IMS]% exit
<< SYSTEM SHUTDOWN >>
```

Then eject T-Kernel 2/x86 Evaluation Kit CD-ROM, and reboot the target.

- **In case of virtual environment (VMware) :**
  Run the virtual machine “`T-Kernel_x86.vmx`” in ”`My Documents\My Virtual Machines\tkx86_vm`”. It is recommended to create its shortcut for later use.
- **In case of real environment :**
  Supply power again to the target machine to reboot.

The display of the target machine (or VMware Player’s window in case of virtual environment) shows an initial window on PMC T-Shell. (Figure 1.12)

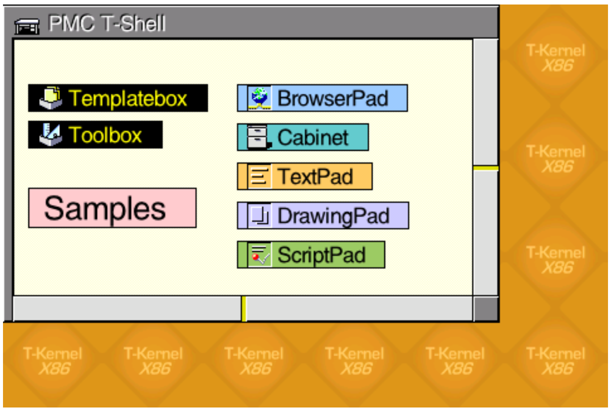


Figure 1.12 An initial window on PMC T-Shell

The terminal software of the host PC shows booting messages of T-Kernel 2/x86, and a prompt “`[/SYS]%`”.

Use `df` command to check the device name of the boot disk (`/SYS`).

```
[/SYS]% df ↩                                          — Check device name
PATH DEV     TOTAL    FREE USED UNIT MAXFILE NAME
/SYS hda0   265040  239752   9% 4096   22528 SYSTEM
```

(14) Screen Size and Colors

The default screen resolution is XGA (1024 × 768), 65536 colors. You can change it by `VIDEOMODE` entry of device configuration file.

```
[/SYS]% devconf VIDEOMODE 2 ↩                  — 640 × 480, 65536 colors
+ 0: VIDEOMODE 2
[/SYS]% exit ↩                                                — Exit CLI
<< EXIT cli >>
[IMS]% exit      1
[IMS]% exit -3 ↩                                         — System reboot
<< SYSTEM REBOOT >>
```

† Refer to *“T-Kernel 2/x86 Implementation Specification”* at “4.6 Screen (Display) Driver” for correspondence between screen modes and `VIDEOMODE` values.

† Instead of `devconf` command, System Setup Utility is also available. Select Right-click → “Utility” → “System Setup” → “Screen size/Color” to change screen mode. The change becomes valid after system reboot.

(15) Additional Font Installation

The "`common\font`" folder of the T-Kernel 2/x86 Evaluation Kit CD-ROM contains many additional fonts, including several English fonts, as well as large sets of Chinese/Japanese/Korean ideographs.

The following install files are for English fonts:

File Name	Font Name
`SSAVALON`	SSAvalon
`SSCOURIE`	SSCourier
`SSEUROMD`	SSEuromode
`SSHELVET`	SSHelvetica
`SSHELVTN`	SSHelveticaN
`SSTIMES`	SSTimes

At least SSHelvetica is recommended to be installed, since T-Shell GUI applications assume it.

Insert T-Kernel 2/x86 Evaluation Kit CD-ROM to the CD-ROM drive of the target machine. In case of virtual environment (VMware), click on "CD/DVD(IDE)" icon on the bottom side of the VMware Player's window and select "connect".

```
[/SYS]% att -c hdc0 CD ↩                      — Attach CD-ROM (partition #0)
hdc0 -> CD (ReadOnly)
[/SYS]% cd /CD/COMMON/FONT ↩                      — Change current directory
[/CD/COMMON/FONT]% INSTFONT SSHELVET ↩                          — Install font
    Font File Name     : regist
 0: **-****            : use
 1: ****-SSHelvetica** : use
```

Use "`exit`" command twice to shutdown the target.

```
[/CD/COMMON/FONT]% exit ↩                                       — Exit CLI
<< EXIT cli >>
[IMS]% exit     1
[IMS]% exit ↩                                            — System shutdown
<< SYSTEM SHUTDOWN >>
```

Then eject the CD-ROM, and reboot the target.

(16) Shutdown the Target System

To shutdown the target system, on the initial window select Right-click → Exit, and click on the Exit button of the confirm dialog.

But the system is still alive in case of debug mode. Then on the terminal software of the host PC, use "`exit`" command twice to shutdown the target.

```
[/SYS]% exit ↩                                   — Exit CLI
<< EXIT cli >>
[IMS]% $logon E
pid = 13 (pri = -1)
[IMS]% exit 1
[IMS]% exit ↩                               — System shutdown
<< SYSTEM SHUTDOWN >>
```

(17) Exit Terminal Software

Exit Terminal Software (Tera Term Pro, HyperTerminal etc.)

† If you install Eclipse Environment in the next chapter, you can use a terminal software "`gterm`" on Eclipse instead of HyperTerminal.

2 Installing Development Environment

2.1 Installing Cygwin

Cygwin is required for the T-Kernel 2/x86 Development Environment on PC.

For more details please also refer to the *"Cygwin Install Manual"* in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

† Windows Login account must be administrator to install Cygwin. And the account name must be in alphanumeric characters without spaces to avoid trouble on Cygwin.

† Even if you have already installed Cygwin, please check its version and installed modules. Refer to *"Cygwin Install Manual"* again and reinstall Cygwin if necessary.

(1) Expand Cygwin Package

Copy (save) the following file to the root of your install drive (such as `C:\`) from the common software directory (`common\soft`) of T-Kernel 2/x86 Evaluation Kit CD-ROM.

Cygwin System Package	`cygwin_d-1.7.17-1.zip`

Expand this zip archive by Windows Explorer.

† Version numbers may be different.

(2) Start the Installer

The installer `setup.exe` is in the expanded folder. Double-click on `setup.exe` to launch the installer program. (Figure 2.1)

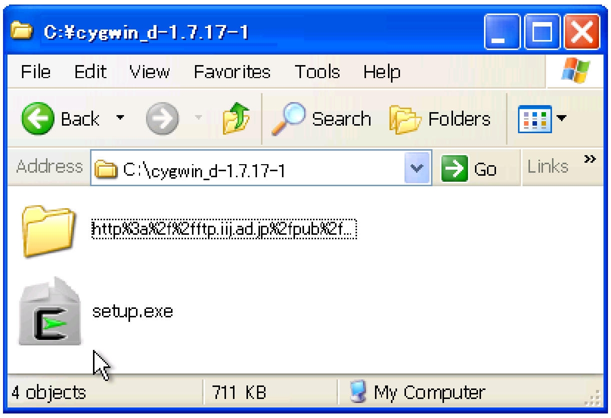


Figure 2.1 Start Cygwin Installer

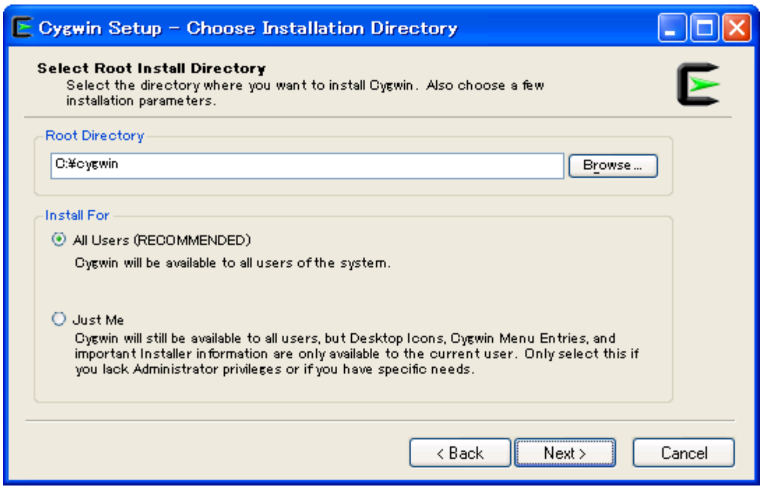
Click on Next.

(3) Select Install Type

Select “Install from Local Directory” and click on Next.

(4) Select Install Directory

Specify where the root directory in Cygwin is to be located in Windows. By default this is `C:\cygwin`. Unless there is some reason for changing this, it can be left as is. (Figure 2.2)



Cygwin Setup - Choose Installation Directory

Select Root Install Directory

Select the directory where you want to install Cygwin. Also choose a few installation parameters.

Root Directory: C:¥cygwin

Install For

● All Users (RECOMMENDED)

Cygwin will be available to all users of the system.

○ Just Me

Cygwin will still be available to all users, but Desktop Icons, Cygwin Menu Entries, and important Installer information are only available to the current user. Only select this if you lack Administrator privileges or if you have specific needs.

Figure 2.2 Select Install Directory

Select “All Users” at “Install For”. It is also the default choice.

And click on Next.

(5) Select Local Package Directory

Specify the folder, where you expanded the Cygwin system package.

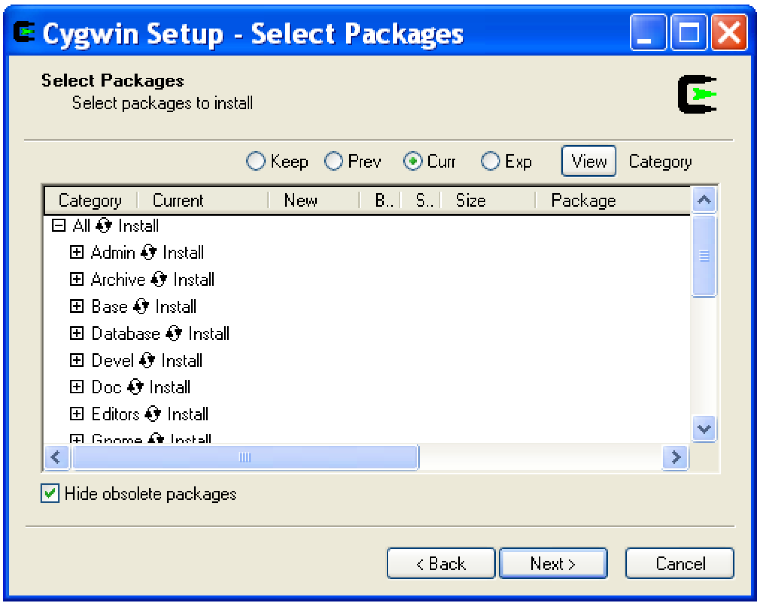
Click on Next to continue.

(6) Select Install Packages

After a file check is performed, a screen for selecting packages to install is displayed.

† A “Setup Alert” dialog appears for the first installation of Cygwin 1.7. Please read it carefully before continuing installation. If you use old version of Cygwin, it will cause some trouble.

The initial selected state is “All ○ Default”. Click on “Default” to change to “Install”. (Figure 2.3)



⊟ All ⟳ Install
⊞ Admin ⟳ Install
⊞ Archive ⟳ Install
⊞ Base ⟳ Install
⊞ Database ⟳ Install
⊞ Devel ⟳ Install
⊞ Doc ⟳ Install
⊞ Editors ⟳ Install
⊞ Gnome ⟳ Install

Figure 2.3 Select Install Package

(7) Install

Then click on Next to proceed with the installation.

(8) Create Icon

Finally a dialog appears asking whether to add icons to the Start menu and desktop. Choose as appropriate, and click on Finish.

If appears “Program Compatibility Assistant”, click on “This program installed correctly”.

(9) Initialize home directory

After modifying the batch file, double-click on the “Cygwin Terminal” icon on the desktop to launch Cygwin. The first time it is started, the user’s home directory is initialized.

(10) Create a symbolic link for gmake

On Cygwin, create a symbolic link from `gmake` to `make`.

```
$ cd /usr/bin ↩
$ ln -s make gmake ↩
```

(11) Create a symbolic link for Perl

On Cygwin, create a symbolic link from `/usr/local/bin/perl` to `/usr/bin/perl`.

```
$ cd /usr/local/bin ↩
$ ln -s /usr/bin/perl ↩
```

Use `exit` command to exit Cygwin.

(12) Set `ja_JP.eucJP` character set

After that log on Windows by the the actual user, double-click on the Cygwin icon on the desktop to launch Cygwin. The first time it is started, the user's home directory is initialized.

Add a line "`LANG=ja_JP.eucJP`" to the `.bashrc` file on your home directory.
For example,

```
$ cd ↩
$ echo LANG=ja_JP.eucJP >> .bashrc ↩
```

On the Cygwin window, click on the mouse right button to show a menu, and select "Option". Select "Text", set "Locale" as "`ja_JP`", and set "Character set" as `eucJP`. Click on OK.

Use `exit` command to exit Cygwin.

2.2 Installing Eclipse

Install Eclipse, an integrated development environment (IDE).

For more details please also refer to the *"Eclipse Install Manual"* in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

(1) Installing Java Runtime

Java Runtime is necessary for running Eclipse.

If you does not installed the Java Runtime, or your Java Runtime is old, please visit the website `http://www.java.com/` to download Java Runtime.

† In case of 64 bit Windows, 32 bit Java Runtime is required. Please install 32 bit Java Runtime.

(2) Expand Eclipse System Package

Copy (save) the following file to a temporary directory of your install drive (such as `C:\temp`) from the common software directory (`common\soft`) of the T-Kernel 2/x86 Evaluation Kit CD-ROM.

Eclipse+CDT	`eclipse-cpp-indigo-SR2-incubation-win32.zip`

† Version numbers may be different.

Expand the archive files above to the root directory of your install drive (such as `C:\`).

† If you expand the packages to `C:\`, the new folder `C:\eclipse` is created, and the packages are expanded under this folder.

(3) Making a shortcut

Make a shortcut on the desktop to the `C:\eclipse\eclipse.exe`.

2.3 Installing T-Kernel Development Environment

Add plug-ins for T-Kernel Development Environment to Eclipse.

For more details please also refer to the *"Eclipse-based T-Kernel Development Environment Install Manual"* and the *"GNU Development Environment (on Eclipse) Manual"* in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

(1) Expand Eclipse plug-ins

Copy (save) the following files to a temporary directory of your install drive (such as `C:\temp`) from the English software directory (`en\soft`) of the T-Kernel 2/x86 Evaluation Kit CD-ROM.

Common part for Eclipse platform	`com.t_engine4u.te.1.2.0.zip`
Machine-dependent part	`com.t_engine4u.tl.pcat.1.2.0.zip`

And expand these archives to the Eclipse directory (such as `C:\eclipse`) by Windows Explorer.

† Please use Windows Explorer to expand these ZIP files. Other software may not correctly expand these archives.

† Version numbers may be different.

(2) Expand T-Shell Additional Package

Copy (save) the following file to a temporary directory of your install drive (such as `C:\temp`) from the English software directory (`en\soft`) of the T-Kernel 2/x86 Evaluation Kit CD-ROM.

T-Shell additional package	`com.t_engine4u.tsh.pcat.1.2.0.zip`

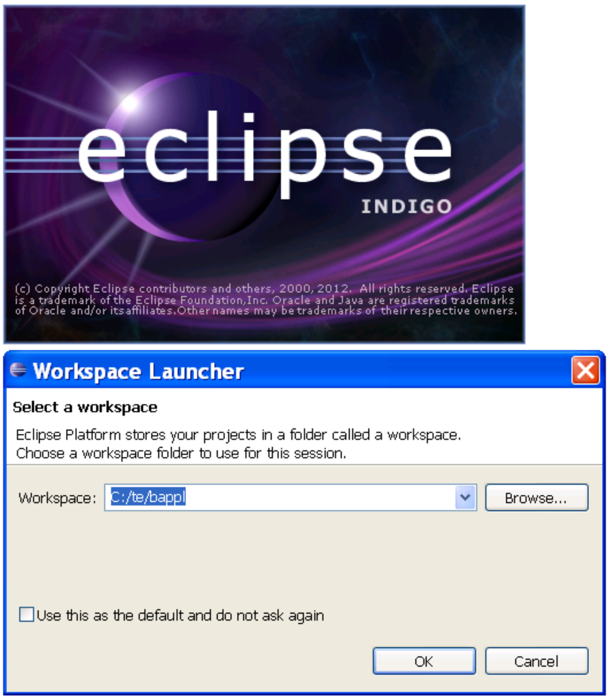
And expand this archive to `C:\eclipse\plugins\com.t_engine4u.tl.pcat.1.2.0_1.2.0\te` by Windows Explorer.

† Version numbers may be different.

† `bappl\.templates` is replaced.

(3) Start Eclipse

Double-click on the shortcut of Eclipse to launch Eclipse (Figure 2.4).



(c) Copyright Eclipse contributors and others, 2000, 2012. All rights reserved. Eclipse is a trademark of the Eclipse Foundation,Inc. Oracle and Java are registered trademarks of Oracle and/or itsaffiliates.Other names may be trademarks of their respective owners.

Select a workspace

Eclipse Platform stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:/te/bappl

□ Use this as the default and do not ask again

Figure 2.4 Starting Eclipse

[**Trouble Shooting**] If appears an error message about Java Runtime Environment, please confirm again about Java Runtime.

(4) Select (or create) a workspace

Appears a dialog of "Select a workspace". Select (or newly create) the workspace `C:\te\bappl` for process-based programs, and click on OK.

† Here a workspace means a directory to save projects. The default workspaces are as follows.

`C:\te\bappl`	Process-based programs
`C:\te\kappl`	T-Kernel-based programs
`C:\te\driver`	Device drivers

(5) Go to the Workbench

Then appears the "Welcome" view. Click on the icon of "Go to the workbench". (Figure 2.5)

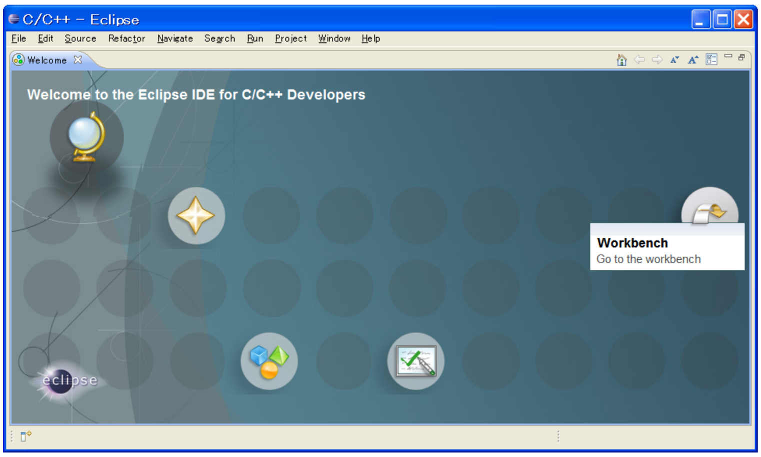

C/C++ – Eclipse
File Edit Source Refactor Navigate Search Run Project Window Help
Welcome
Welcome to the Eclipse IDE for C/C++ Developers
Workbench
Go to the workbench
eclipse

Figure 2.5 "Welcome" view of Eclipse

(6) Open the "T-Engine Development" Perspective

Select Toolbar → "Window" → "Open Perspective" → "Other...'.
Select "T-Engine Development", and click on OK. (Figure 2.6)

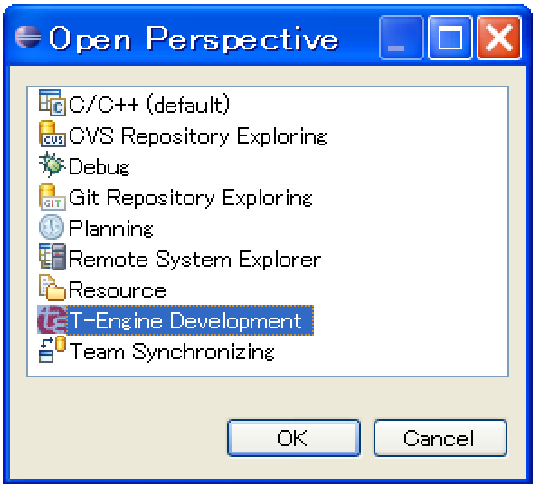
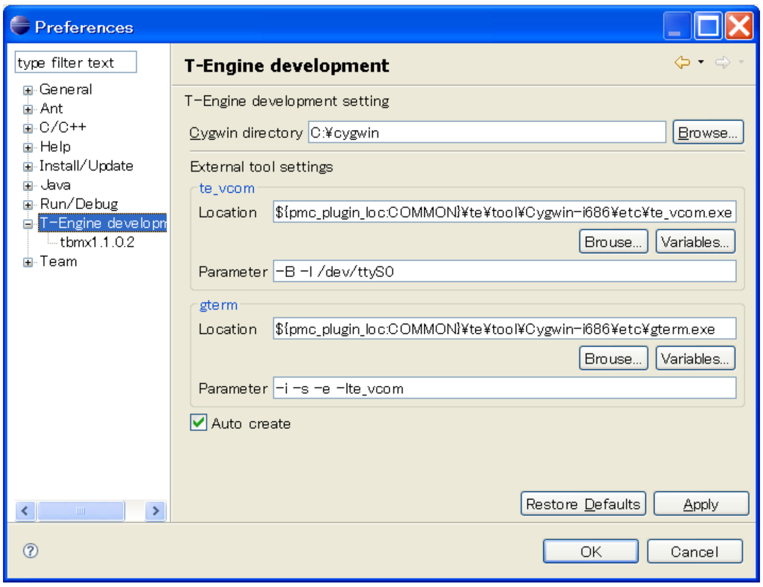


Figure 2.6 Select T-Engine development perspective

(7) Setting of T-Engine Development

Select Toolbar → “Window” → “Preferences” to open the Preferences Dialog. Select the page of “T-Engine Development”. (Figure 2.7)



T-Engine development

T-Engine development setting

Cygwin directory C:¥cygwin

External tool settings

te_vcom

Location ${pmc_plugin_loc:COMMON}¥te¥tool¥Cygwin-i686¥etc¥te_vcom.exe

Parameter -B -l /dev/ttyS0

gterm

Location ${pmc_plugin_loc:COMMON}¥te¥tool¥Cygwin-i686¥etc¥gterm.exe

Parameter -i -s -e -lte_vcom

☑ Auto create

Figure 2.7 T-Engine Development Settings

As the parameter of Cygwin directory, specify your Cygwin installed directory, like `C:\cygwin`.

As the parameter of `te_vcom`, specify as follows:

- **In case of virtual environment (VMware) :**

  As the parameter of `te_vcom`, specify "`-l localhost`".

  † `-l` is minus ell.

- **In case of real environment :**

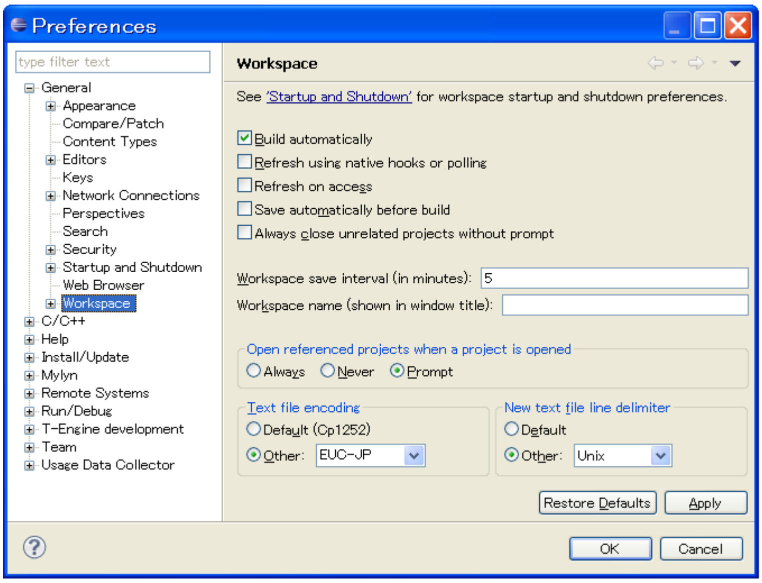
  As the parameter of `te_vcom`, specify "`-B -l /dev/ttyS0`" in case of COM port number COM1, "`-B -l /dev/ttyS1`" in case of COM2, and so on.

  † "`-B`" means 115200 bps, "`-l`" is minus ell, which stands for line device, and "`/dev/ttyS`n" means COM($n + 1$).

(8) Setting of Workspace

Select the page of "General" → "Workspace". (Figure 2.8)

- "Build automatically" is to be checked.
- "Text file encoding" is to be Other → `EUC-JP`.
- "New text file line delimiter" is to be Other → `Unix`.



Workspace

See 'Startup and Shutdown' for workspace startup and shutdown preferences.

☑ Build automatically
☐ Refresh using native hooks or polling
☐ Refresh on access
☐ Save automatically before build
☐ Always close unrelated projects without prompt

Workspace save interval (in minutes): 5
Workspace name (shown in window title):

Open referenced projects when a project is opened
○ Always ○ Never ● Prompt

Text file encoding
○ Default (Cp1252)
● Other: EUC-JP

New text file line delimiter
○ Default
● Other: Unix

Figure 2.8 Setting of Workspace

And click on OK to close the Preference Dialog.

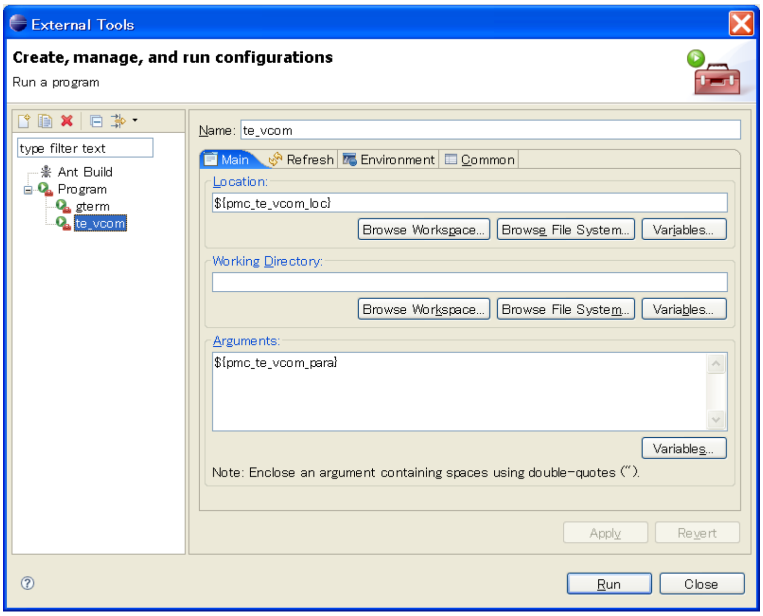
This configuration must be done for each workspace (`C:\te\bappl`, `C:\te\kappl`, `C:\te\driver` etc.).

(9) Opening console

Select Toolbar → "External Tools" → "External Tools Configurations..." to show the "External Tools Configurations" Dialog.

On the dialog, click on the "+" at the left of "Program". Then appear `te_vcom` and `gterm` under "Program".

Select `te_vcom` and click on the Run button. (Figure 2.9)



Name: te_vcom

Location: ${pmc_te_vcom_loc}

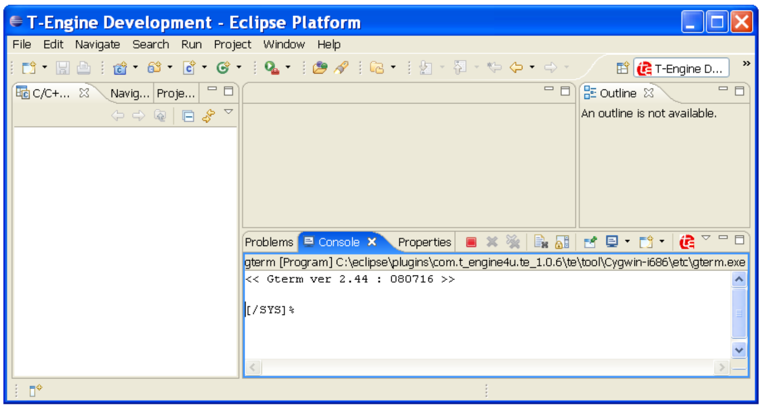
Arguments: ${pmc_te_vcom_para}

Note: Enclose an argument containing spaces using double-quotes (").

Figure 2.9 Running `te_vcom`

Open the "External Tools Configurations" Dialog again, select `gterm`, and click on the Run button.

On the console view of `gterm`, hit ↩ (Enter) key several times. Then appears a prompt (such as `[/SYS]%`) from the target side (T-Kernel 2/x86). This is a prompt of CLI (Command Line Interpreter), which runs on T-Kernel 2/x86. (Figure 2.10)


T-Engine Development - Eclipse Platform
File Edit Navigate Search Run Project Window Help
T-Engine D...
C/C+... Navig... Proje...
Outline
An outline is not available.
Problems Console Properties

gterm [Program] C:\eclipse\plugins\com.t_engine4u.te_1.0.6\te\tool\Cygwin-i686\etc\gterm.exe

```
<< Gterm ver 2.44 : 080716 >>

[/SYS]%
```

Figure 2.10 CLI prompt on `gterm`

Next time, the shortcuts to these external tools (`te_vcom` and `gterm`) will appear on the pull-down menu of External Tools.

3 Development

3.1 Starting Target and Development Environment

Start T-Kernel 2/x86 and Eclipse as follows.

(1) Boot the target side (T-Kernel 2/x86)

- **In case of virtual environment (VMware) :**

  Run the virtual machine "`T-Kernel_x86.vmx`" in "`My Documents\My Virtual Machines\tkx86_vm`". It is recommended to create its shortcut for later use.

  VMware Player's window shows an initial window on PMC T-Shell. (Figure 1.12 in page 18)

  > [**Trouble Shooting**] An initial window will not appear due to the previous exit situation. In this case try to reset the virtual machine by "Player" → "Troubleshoot" → "Reset" at the topside of the VMware Player's window.

- **In case of real environment :**

  Boot the target machine.

  The display of the target machine shows an initial window on PMC T-Shell.

(2) Start Eclipse

Double-click on the shortcut of Eclipse to launch Eclipse.

Select a workspace, such as `C:\te\bappl` for process-based programs.

(3) Run `te_vcom` and `gterm`

Select the external tools, and run `te_vcom` and `gterm`.

† These external tools do not run just after starting Eclipse. Every time you started Eclipse (or switched workspace), you also have to run `te_vcom` and `gterm`.

On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`) from the target side (T-Kernel 2/x86).

† Do not use other terminal software including HyperTerminal while using `te_vcom` and `gterm`. And do not start `te_vcom` or `gterm` twice.

[**Trouble Shooting**] If CLI prompt does not appear (including T-Monitor prompt `TM>`), please reset the target.

In case of virtual environment (VMware), reset the virtual machine by "Player" → "Troubleshoot" → "Reset" at the topside of the VMware Player's window.

3.2 Console Tools

T-Kernel 2/x86 comes with a variety of console tools useful for development, as well as Eclipse dialogs. By Eclipse dialogs one can transfer and execute programs without console operation. But for other miscellany operations, you can use these console tools on `gterm`.

CLI (Command Line Interpreter)

A process-based command line interpreter, providing powerful features essential for development, especially file operation features. The standard prompt is `[/SYS]%`. See the *"Development Tool Manual"* at "3. Command Line Interpreter (CLI)" for details.

† The CLI prompt shows the working (current) directory name, which is set by `cd` command.

Utilities on CLI

Various utilities, such as "`hdpart`" (partitioning disks) and "`format`" (format disks) are stored in the system disk, and available as external commands on CLI. See the *"Development Tool Manual"* at "4. Utilities", "5. UNIX (File) Emulator Command Tools" and "6. Network Utilities" for details.

IMS (Initial Monitor System)

IMS is a T-Kernel based monitor. Its prompt is `[IMS]%` by default. See the "*Development Tool Manual*" at "2. IMS" for details.

T-Monitor

The most basic monitor program. Its forte is hardware-level operations and the like. The prompt is `TM>`.

T-Monitor commands are listed in the *"T-Monitor Specification"* at 3.3, "List of Commands." T-Kernel 2/x86 adds additional commands to T-Monitor for Flash ROM writing and other functions. These additional commands are given in the *"Implementation Specification"* at 2, "T-Monitor Implementation Specification."

A simple practice is given below to illustrate use of the CLI and T-Monitor.

(1) Confirm CLI Prompt

On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

```
[/SYS]%
```

(2) List of CLI Commands

Enter the CLI command “?” to display a list of CLI commands.

```
[/SYS]% ? ↩
                        CLI command list
att     det     eject   cd      ls      fs  ··· omitted ···
```

(3) CLI Command Help

Use the CLI command “?” to display a description of the “`ls`” command.

```
[/SYS]% ?  ls ↩
ls [-f][-F][-l|-t] [<path name>..]
list files under the path name
··· omitted ···
```

(4) List of Files

Use the CLI command “`ls`” to display a list of files.

```
[/SYS]% ls ↩
SBOOT         KERNEL.SYS    SYSCONF     ··· omitted ···
```

(5) Switch to T-Monitor

Use the CLI command “`#`” to switch to T-Monitor.

```
[/SYS]% # ↩
TM>
```

(6) List of T-Monitor Commands

Use the T-Monitor command “?” to display a list of T-Monitor commands.

```
TM> ? ↩
--- Command List :   "? command" for details ---
DumpByte/Half/Word(D/DB/DH/DW)  ··· omitted ···
```

(7) Show Registers

Use the T-Monitor command "R" (Register) to show CPU registers.

```
TM> r ↩
EIP : 001117D7  EFLAGS: 00001246 IF:1 ··· omitted ···
```

(8) Return from T-Monitor

Use the T-Monitor command "g" (Go) to return to the CLI.

```
TM> g ↩
[/SYS]%
```

3.3 Process-Based and T-Kernel-Based Programs

Programs that run on T-Kernel 2/x86 are generally classified as either process-based programs or T-Kernel-based programs. (Additionally there are monitor-based programs, but these are not dealt with here.)

Process-based programs :

These are programs that run on T-Kernel Extension.

Ordinary application programs are generally of this type.

And process runs in memory protection mode for system safety such that process failure will not cause system down.

As memory space, an independent local space is assigned to each process. A process has one or more tasks, which shares the local space of the process.

A process is run by invoking the `main()` function, and when the `main()` function terminates, the process terminates as well.

APIs available in process-based programs are described in the following documents in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

- "*T-Kernel Extension Manual*"
  Process, tasks, files, events and so on.
- "*PMC T-Shell Manual*"
  Screen drawing, fonts, GUI and network (TCP/IP) functions.

  † Basically PMC T-Shell is available only on process.

- “*Library Manual*”

  Standard C library and other libraries, including `fopen()` and other `<stdio.h>` functions.

- “*Device Driver Manual*”

  Process-based programs can call device drivers.

T-Kernel-based programs :

These are programs that make direct use of T-Kernel features.

This type of programming is geared to creating device drivers and the like, since hardware control and interrupts can be handled. As memory space, multiple T-Kernel-based programs are loaded into the same system space and are made resident.

A T-Kernel-based program is loaded into system space by the `lodspg` command. At this time the `main()` function is called, and the first argument `ac` indicates the number of arguments passed. Ordinarily, task and handler creation and other initialization takes place in this processing.

The `unlspg` command can be used to unload a program. Here, the `main()` function is called again; but this time a minus value is passed in `ac`. Post-processing such as deleting the initial task and handlers normally takes place in this processing.

APIs available in T-Kernel-based programs are described in the following documents in the T-Kernel 2/x86 Evaluation Kit CD-ROM.

- “*T-Kernel Specification*”

  T-Kernel/OS provides the kernel functions of real time OS, such as tasks, semaphores, interrupt handlers. And T-Kernel/SM provides device driver management and so on.

- “*Library Manual*”

  Standard C library and other libraries.

  † Some library are process-based only, or T-Kernel-based only, but other library can be commonly used.

- “*Device Driver Manual*”

  T-Kernel-based programs can call device drivers, as well as making device drivers.

3.4 "Hello, world" as a Process-Based Program

First of all, we'll try the well-known program "Hello, world" as a process-based program. For more details refer to "*GNU Development Environment (on Eclipse) Manual*".

(1) Select a Workspace

Launch Eclipse, and on the "Select a workspace" dialog, select (or newly create) the workspace "`C:\te\bappl`" for process-based programs, and click on OK.

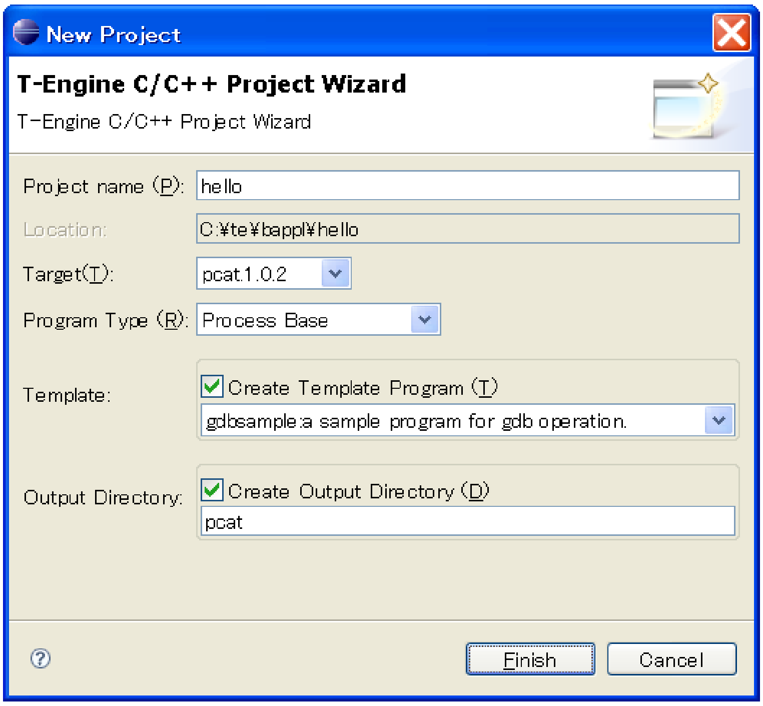
If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\bappl`", by Toolbar → "File" → "Switch Workspace...".

† If you switched workspace, also terminate external tools `te_vcom` and `gterm`. Then you have to run `te_vcom` and `gterm` again. On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(2) Create a Project

On the "C/C++ Project" View, click on the mouse right button to show a menu, and select "New" → "T-Engine C/C++ Project". (Or select Toolbar → "New" → "T-Engine C/C++ Project".)

Specify the following items. (Figure 3.1)



New Project

T-Engine C/C++ Project Wizard

T-Engine C/C++ Project Wizard

Project name (P): hello

Location: C:¥te¥bappl¥hello

Target(T): pcat1.0.2

Program Type (R): Process Base

Template: ☑ Create Template Program (T)
gdbsample:a sample program for gdb operation.

Output Directory: ☑ Create Output Directory (D)
pcat

Figure 3.1 New Project Dialog

- Project name :
  Specify your project name, for example, `hello`.
- Target :
  Specify your target machine, if you have already installed two or more development environments.
- Program Type:
  Specify "Process Base", or "Decide by workspace", if your workspace is `C:\te\bappl`.
- Template:
  Check it and select the template "gdbsample: a sample program for gdb operation".
- Output Directory:
  Check it.

And click on the Finish button to create a new project.

(3) Create a source program

The sources of the template are copied under the src directory of the project hello in the "C/C++ Project" View. Double-click on hello to view the source files.

Here modify these source files as follows to make "Hello, world".

- hello/src/main.c :

  Modify the main function as follows:

```
/* Hello, world (process-based) */
#include <basic.h>     /* basic common header */
#include <stdio.h>     /* printf() etc. */
W main( W ac, TC *av[] )
{
    printf( "Hello, world\n" );
    return 0;
}
```

  Save the modified file by Toolbar → "File" → "Save".

  † Here the types W and TC are defined by T-Kernel Specification. And return 0 returns the exit code of process.

- hello/src/Makefile :

  Change the make target to "TARGET = hello".

  Change the list of source files to "SRC = main.c". (Delete sub.c.)

  Save the modified file by Toolbar → "File" → "Save".

- hello/src/sub.c, hello/src/sub.h :

  Delete these files by right-click menu.

(4) Make (compile and link)

Select the target Makefile, hello/pcat/Makefile, in the "C/C++ Project" View.

And then, select Toolbar → "Project" → "Make T-Engine Target all" to make (compile and link) your program.

If the make is succeeded, the executable file hello/pcat/hello is generated (with a ▷ icon).

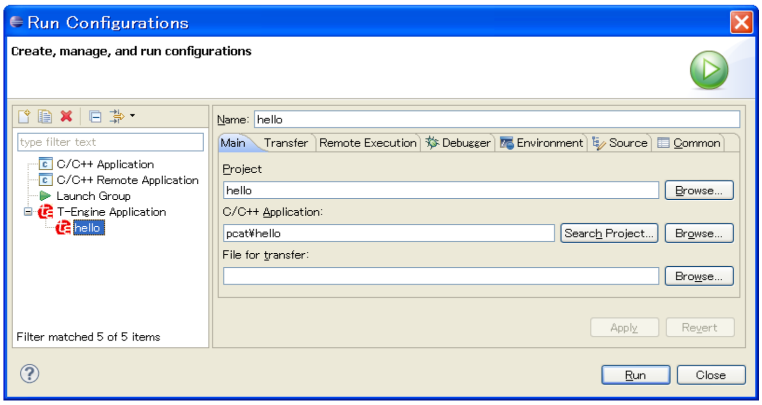
[Trouble Shooting] If appears a message "Error launching builder", confirm that you have selected the make target hello/pcat/Makefile.

[Trouble Shooting] If appears a message "makedeps: Command not found", confirm installation of Perl on `/usr/local/bin/perl` after Cygwin installation.

(5) Transfer and Execute the Executable File

Select the executable file `hello/pcat/hello`, click on the mouse right button to show a menu, and select "Run As" → "Run Configurations...". Then appears the Run Configurations Dialog.

On the dialog, select "T-Engine Application", click on the mouse right button to show a menu, and select "New". Then appears the default settings to transfer and execute the program. (Figure 3.2)


Run Configurations

Create, manage, and run configurations

- C/C++ Application
- C/C++ Remote Application
- Launch Group
- T-Engine Application
  - hello

Filter matched 5 of 5 items

Name: hello

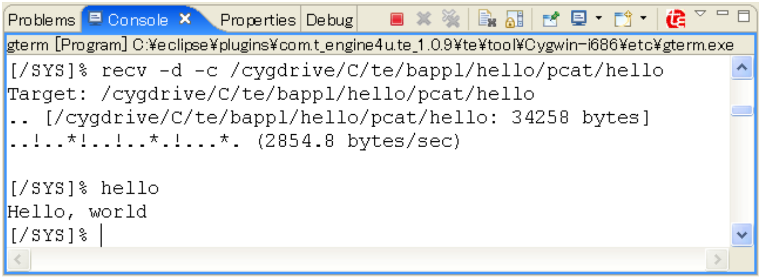
Project: hello

C/C++ Application: pcat¥hello

File for transfer:

Figure 3.2 Run Dialog

Click on the Run button at the bottom of the dialog to transfer and execute the program. Then the process is executed, and appears "Hello, world" on the `gterm` console. (Figure 3.3)


gterm [Program] C:¥eclipse¥plugins¥com.t_engine4u.te_1.0.9¥te¥tool¥Cygwin-i686¥etc¥gterm.exe

```
[/SYS]% recv -d -c /cygdrive/C/te/bappl/hello/pcat/hello
Target: /cygdrive/C/te/bappl/hello/pcat/hello
.. [/cygdrive/C/te/bappl/hello/pcat/hello: 34258 bytes]
..!..*!..!..*.!...*. (2854.8 bytes/sec)

[/SYS]% hello
Hello, world
[/SYS]% 
```

Figure 3.3 "Hello, world" appears on console

[**Trouble Shooting**] The serial communication is required to transfer files to the target.

Confirm that you have run `te_vcom` and `gterm` again, and on the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

[**Trouble Shooting**] If appears a message "`Can't Create (-1966080)`", it means read only error. Please check that T-Kernel 2/x86 is booted from writable disk.

3.5 "Hello, world" as a T-Kernel-Based Program

Next we'll try the "Hello, world" as a T-Kernel-based program. It puts "Hello, world" at loading, and "See you again" at unloading.

(1) Select a Workspace

Launch Eclipse, and on the "Select a workspace" dialog, select (or newly create) the workspace "`C:\te\kappl`" for T-Kernel-based programs, and click on OK.

If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\kappl`", by Toolbar → "File" → "Switch Workspace...".

† If you switched workspace, also terminate external tools `te_vcom` and `gterm`. Then you have to run `te_vcom` and `gterm` again. On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(2) Create a Project

On the "C/C++ Project" View, click on the mouse right button to show a menu, and select "New" → "T-Engine C/C++ Project". (Or select Toolbar → "New" → "T-Engine C/C++ Project".)

Specify the following items.

- Project name :
  Specify your project name, for example, `hello2`.
- Target :
  Specify your target machine, if you have already installed two or more development environments.
- Program Type:
  Specify "T-Kernel Base", or "Decide by workspace", if your workspace is `C:\te\kappl`.

- Template:
  Check it and select the template “sample: print strings sample program”.
- Output Directory:
  Check it.

And click on the Finish button to create a new project.

(3) Create a source program

The sources of the template are copied under the `src` directory of the project `hello2` in the “C/C++ Project” View. Double-click on `hello2` to view the source files.

Here modify these source files as follows to make “Hello, world”.

- `hello2/src/main.c` :
  Modify the main function as follows:

```
/* Hello, world (T-Kernel-based) */
#include <basic.h>       /* basic common header */
#include <tk/tkernel.h>  /* T-Kernel header */
#include <stdio.h>       /* printf() etc. */

ER main( INT ac, UB *av[] )
{
    if (ac >= 0) {
        printf("Hello, world\n");
    } else {
        printf("See you again\n");
    }
    return E_OK;
}
```

  Save the modified file by Toolbar → “File” → “Save”.
- `hello2/src/Makefile` :
  Change the make target to “`TARGET = hello2`”.
  Save the modified file by Toolbar → “File” → “Save”.

(4) Make (compile and link)

Select the target Makefile, `hello2/pcat/Makefile`, in the “C/C++ Project” View.

And then, select Toolbar → "Project" → "Make T-Engine Target all" to make (compile and link) your program.

If the make is succeeded, the relocatable executable file `hello2/pcat/hello2` is generated (without a ▷ icon, but it is correct). And `hello2.abs` is a absolute address executable file, and `hello2.trg` is its stripped file.

(5) Transfer and Execute the Executable File

Select the executable file `hello2/pcat/hello2`, click on the mouse right button to show a menu, and select "Run As" → "Run Conrigurations ...". Then appears the Run Configurations Dialog.

On the dialog, select "T-Engine Application", click on the mouse right button to show a menu, and select "New". Then appears the default settings to transfer and execute the program.

Click on the Run button at the bottom of the dialog to transfer and execute the program. Then the program is loaded on the system common space, and appears "Hello, world" on the `gterm` console.

(6) Show the System Common Space

On the `gterm` console , use `ref spg` command to confirm that `hello2` is loaded.

```
[/SYS]% ref spg ↩
[ 1] - 0xc0261000 -  5  vmwdrv
··· omitted ···
[14] - 0xc056b000 -  4  hello2
[/SYS]%
```

† System program ID and load address may be different.

(7) Unload the program

Now unload `hello2` by executing the "`unlspg`" command, specifying the program by its system program ID as displayed in the previous step (14 in the above example). Then appears "Hello, world" on the `gterm` console.

```
[/SYS]% unlspg 14 ↩
See you again
[/SYS]%
```

4 Running the Sample Programs

The Tutorial page on the T-Kernel 2/x86 Evaluation Kit CD-ROM contains a collection of sample programs.

- **Process application samples**

  (1) Juggling
  This program shows juggling animation.

  (2) Tiny Web Server
  This program transfers files from T-Kernel 2/x86 to Web browsers on other PCs by TCP/IP network.

- **T-Kernel-based samples on T-Kernel 2.0 new functions**

  T-Kernel 2.0 specification defines the following new functions as standards:
  · Time control in microseconds
  · Physical timer
  · Large capacity storage device support
  · Cache control, memory access right
  · Fast lock

  The T-Kernel 2/x86 Evaluation Kit CD-ROM contains the following samples.

  (3) Time-in-microseconds sample
  This is a sample program using `tk_get_otm_u` (get operating time in microseconds) and `tk_cre_cyc_u` (create cyclic handler in microseconds).

  (4) Physical timer sample implementation
  This sample program uses physical timer to measure time to start task (`tk_sta_tsk`).

  > There are several methods and implementations for physical timer, depending on hardware and purpose. Hence T-Kernel 2/x86 does not implement physical timer as a standard, while one sample implementation is given in a source program. Refer to the header comment of the source program about specification and limitation of this sample physical timer.

4.1 Juggling

This process-based program shows juggling animation, using PMC T-Shell drawing functions.

This sample is also useful to learn debug operations: set break points, change variables and so on.

(1) Select a Workspace

Launch Eclipse, and on the “Select a workspace” dialog, select the workspace “`C:\te\bappl`” for process-based programs, and click on OK.

If you have already launched Eclipse and works on other workspace, switch workspace to “`C:\te\bappl`”, by Toolbar → “File” → “Switch Workspace...”.

† If you switched workspace, also terminate external tools `te_vcom` and `gterm`. Then you have to run `te_vcom` and `gterm` again. On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(2) Create a Project

On the “C/C++ Project” View, click on the mouse right button to show a menu, and select “New” → “T-Engine C/C++ Project”. (Or select Toolbar → “New” → “T-Engine C/C++ Project”.)

Specify the following items.

- Project name :
  Enter the project name, `juggling`.
- Target :
  Specify your target machine.
- Program Type:
  Specify “Process Base”, or “Decide by workspace”, if your workspace is `C:\te\bappl`.
- Template:
  Do not check it, because the source program will be expanded later.
- Output Directory:
  Check it.

And click on the Finish button to create a new project.

(3) Expand Source Program

The source program is archived in “`juggling.zip`” on the “`en\tutorial`” folder in the T-Kernel 2/x86 Evaluation Kit CD-ROM. Expand this archive “`juggling.zip`” to “`C:\te\bappl`” by Windows Explorer.

In the “C/C++ Project” View, select right-click menu → “Refresh”. Now the source files appear under the “`src`” directory of the project “`juggling`” in the “C/C++ Project” View. Double-click on “`juggling`” to view the source files.

(4) Make (compile and link)

Select the target Makefile, "`juggling/pcat/Makefile`", in the "C/C++ Project" View.

And then, select Toolbar → "Project" → "Make T-Engine Target all" to make (compile and link) your program.

If the make is succeeded, the executable file "`juggling/pcat/juggling`" is generated (with a ▷ icon).

(5) Transfer and Execute the Executable File

Select the executable file "`juggling/pcat/juggling`", click on the mouse right button to show a menu, and select "Run As" → "Run Configurations...". Then appears the Run Configurations Dialog.

On the dialog, select "T-Engine Application", click on the mouse right button to show a menu, and select "New". Then appears the default settings to transfer and execute the program.

Click on the Run button at the bottom of the dialog to transfer and execute the program. Juggling animation is showed on the screen (Figure 4.1)

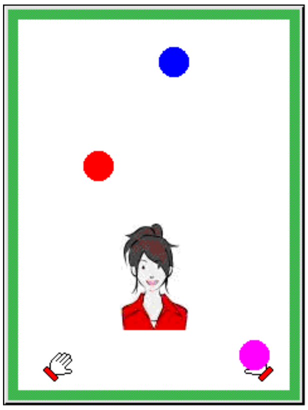


Figure 4.1 Run Juggling

(6) Make for Debugging

To debug your program, the executable file must include debug information. To make that file, make your program on the folder with a debug option, such as "`pcat.debug`".

Select the target Makefile, "`juggling/pcat.debug/Makefile`", in the "C/C++ Project" View. And then, select Toolbar → "Project" → "Make T-Engine Target all" to make (compile and link) your program.

If the make is succeeded, the executable file "`juggling/pcat.debug/juggling`" is generated with debug information (with a bug icon).

(7) Start Debugging

Select the executable file "`juggling/pcat.debug/juggling`", click on the mouse right button to show a menu, and select "Debug As" → "Debug Configurations...". Then appears the Debug Configurations Dialog.

On the dialog, select "T-Engine Application", click on the mouse right button to show a menu, and select "New". Then appears the default settings to transfer and debug the program.

† There remains a previous setting created in (5) for execution. But create a new one for debugging.

Click on the Debug button at the bottom of the dialog to transfer and debug the program. If appears a "Confirm Perspective Switch" dialog, click on Yes to switch perspective. Then appears the Debug perspective.

[**Trouble Shooting**] If terminates the debugger while its start, you have to extend debugger timeout.

(1) Remove console of the terminated debugger:
Click on the "Remove All Terminated Launches" (double cross) button at the top of the console view.

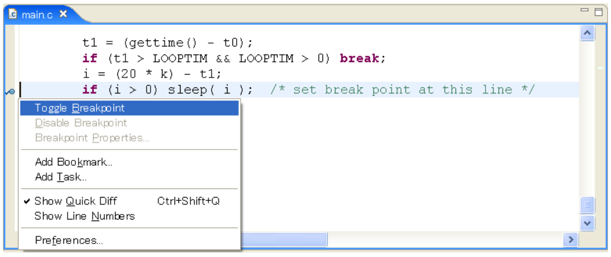
(2) Extend debugger timeout:
Select Toolbar → "Window" → "Preferences", open the page of "C/C++" → "Debug" → "GDB MI", and extend the values of "Debugger timeout" and "Launch timeout".

(3) Start debug again by the Debug Configurations Dialog.

(4) Set a Breakpoint

The program is stopped at the beginning of the `main()` function.

Now set a breakpoint in the line "`if (i > 0) sleep( i );`" near the end of the `main()` function. At the leftmost side of this line, click on the mouse right button to show a menu. Select "Toggle Breakpoint" to set a breakpoint. (Figure 4.2)

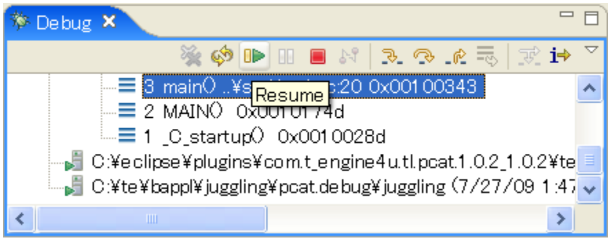


```
        t1 = (gettime() - t0);
        if (t1 > LOOPTIM && LOOPTIM > 0) break;
        i = (20 * k) - t1;
        if (i > 0) sleep( i );  /* set break point at this line */
```

Figure 4.2 Set a Breakpoint

(5) Continue Execution

Click on the “Resume” (green right triangle) button in the debug view to continue execution. The program runs until a breakpoint. (Figure 4.3)



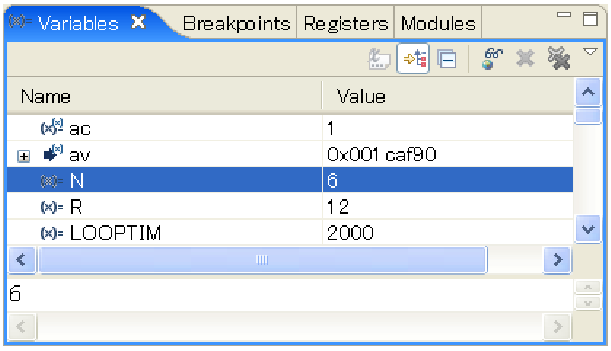
```
3 main() ..¥s        c:20 0x00100343
2 MAIN() 0x0010174d
1 _C_startup() 0x0010028d
C:¥eclipse¥plugins¥com.t_engine4u.tl.pcat.1.0.2_1.0.2¥te
C:¥te¥bappl¥juggling¥pcat.debug¥juggling (7/27/09 1:47
```

Figure 4.3 Continue Execution

By repeating this continuation, the balls move step by step on the target screen.

(6) Change Variables

When the program is stopped at a breakpoint, change the value of the variable N (number of balls) in the variables view from 3 to 6. (Figure 4.4)



Name	Value
ac	1
av	0x001caf90
N	6
R	12
LOOPTIM	2000

6

Figure 4.4 Change a Variable

By continuing execution, the number of balls becomes 6 on the target screen.

In the same manner one can change the value of the variable `R` (radius of balls on the screen).

(7) Terminate Debugging

Click on the "Terminate" (red square) button in the debug view to terminate debugging. And click on the "Remove All Terminated Launches" (double cross) button in the same view.

Select "≫" at the right-top of the Eclipse window to back to the "T-Engine Development" Perspective.

4.2 Tiny Web Server

This process-based tiny web server is provided as an example of a network application using TCP/IP functions of PMC T-Shell. Files on T-Kernel 2/x86 are transferred to web browsers on other PCs.

(1) Select a Workspace

Launch Eclipse, and on the "Select a workspace" dialog, select the workspace "`C:\te\bappl`" for process-based programs, and click on OK.

If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\bappl`", by Toolbar → "File" → "Switch Workspace...".

† If you switched workspace, also terminate external tools `te_vcom` and `gterm`. Then you have to run `te_vcom` and `gterm` again. On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(2) Network Configuration

On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`). Then use `netconf` command for network configuration.

If the network environment is one that assigns IP addresses automatically by DHCP, the default setting of 0.0.0.0 can be left as is. If not, first use the "`netconf c`" command to assign an IP address to T-Kernel 2/x86.

```
[/SYS]% netconf c  ↩
hostname   = ? tkernel ↩                              — local host name
host ip    = 0.0.0.0 ? 192.168.0.70 ↩                 — local IP address
```

```
dns1name   = ? ↩                                  — DNS server #1
dns1 ip    = 0.0.0.0 ? ↩
dns2name   = ? ↩                                  — DNS server #2
dns2 ip    = 0.0.0.0 ? ↩
domain     = ? ↩                                  — domain name
gateway ip = 0.0.0.0 ? ↩                          — gateway
subnetmask = 255.255.255.0 ? ↩                    — subnet mask
wlan       = none (n/a/i)? ↩                      — no wireless LAN
[/SYS]%
```

† Instead of `netconf` command, Network Setup Utility (right-click → "Utility" → "Network Setup") is also available.

(3) Confirm Network Connection

On the console view of `gterm`, use the "`ping`" command to confirm the network connection.

```
[/SYS]% ping localhost ↩                          — ping self
localhost is alive <192.168.0.70> : 0 ms
[/SYS]% ping 192.168.0.1 ↩                        — pint another machine
192.168.0.1 is alive <192.168.0.1> : 0 ms
```

† In this example, the T-Kernel 2/x86 IP address is 192.168.0.70 and that of another machine is 192.168.0.1. Change these as appropriate for the particular network environment.

† In case of virtual environment (VMware), select "Network Adapter" icon at the bottom side of the VMware Player's window, and select "connect". And also select "Settings...", and select "Bridged", "NAT" or "Host-only", depending on your network environment. For example "Bridged" is used to connect the virtual machine directly to network as well as other physical machines.

† In case of real environment, select right-click → "Utility" → "System Setup" → "Network adapter" to set network device.

(4) Transfer Image Files

Transfer Image Files to the target side (T-Kernel 2/x86), for the contents of the tiny web server.

On the console view of `gterm`, use the "`recv`" command to transfer image files from the host PC to the target side (T-Kernel 2/x86).

For example, if you want to transfer `C:\tmp\abc.jpg`,

```
[/SYS]% recv -c -d /cygdrive/c/tmp/abc.jpg ↩
```

Transfer several image files in the same manner (those extension must be `.jpg` or `.png`).

(5) Create a Project

On the "C/C++ Project" View, click on the mouse right button to show a menu, and select "New" → "T-Engine C/C++ Project". (Or select Toolbar → "New" → "T-Engine C/C++ Project".)

Specify the following items.

- Project name :
  Enter the project name, `webserv`.
- Target :
  Specify your target machine.
- Program Type:
  Specify "Process Base", or "Decide by workspace", if your workspace is `C:\te\bappl`.
- Template:
  Do not check it, because the source program will be expanded later.
- Output Directory:
  Check it.

And click on the Finish button to create a new project.

(6) Expand Source Program

The source program is archived in "`webserv.zip`" on the "`en\tutorial`" folder in the T-Kernel 2/x86 Evaluation Kit CD-ROM. Expand this archive "`webserv.zip`" to "`C:\te\bappl`" by Windows Explorer.

In the "C/C++ Project" View, select right-click menu → "Refresh". Now the source files appear under the "`src`" directory of the project "`webserv`" in the "C/C++ Project" View. Double-click on "`webserv`" to view the source files.

(7) Make (compile and link)

Select the target Makefile, `webserv/pcat/Makefile`, in the "C/C++ Project" View.

And then, select Toolbar → "Project" → "Make T-Engine Target all" to make (compile and link) your program.

If the make is succeeded, the executable file "`webserv/pcat/webserv`" is generated (with a ▷ icon).

(8) Transfer and Execute the Executable File

Select the executable file "`webserv/pcat/webserv`", click on the mouse right button to show a menu, and select "Run As" → "Run Configurations ...". Then appears the Run Configurations Dialog.

On the dialog, select "T-Engine Application", click on the mouse right button to show a menu, and select "New". Then appears the default settings to transfer and execute the program.

Click on the Run button at the bottom of the dialog to transfer and execute the program.

(9) View from Web Browser

From another PC on the network, enter the T-Kernel 2/x86 IP address in a Web browser, and confirm that the images loaded earlier are listed in the browser. If, for example, the T-Kernel 2/x86 IP address is 192.168.0.70, enter the URL as follows:

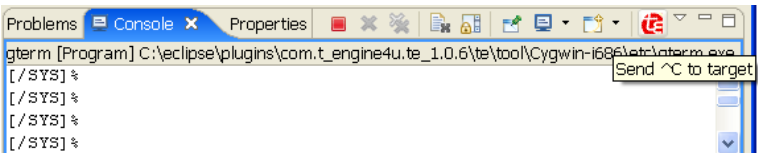
```
http://192.168.0.70/index.html
```

† `index.html` is the image list page, which is generated automatically by the tiny web server.

(10) Terminate Process

On the console, send `^C` (Ctrl-C) to terminate the web server process.

Use the TE logo button upside the console view of Eclipse. (Figure 4.5)



```
gterm [Program] C:\eclipse\plugins\com.t_engine4u.te_1.0.6\te\tool\Cygwin-i686\etc\gterm.exe
[/SYS]%
[/SYS]%
[/SYS]%
[/SYS]%
```

Figure 4.5 Send ^C on Eclipse

4.3 Time-in-microseconds Sample

This is a T-Kernel-based sample program using `tk_get_otm_u` (get operating time in microseconds) and `tk_cre_cyc_u` (create cyclic handler in microseconds).

(1) Select a Workspace

Launch Eclipse, and on the "Select a workspace" dialog, select the workspace "`C:\te\kappl`" for T-Kernel-based programs, and click on OK.

If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\bappl`", by Menubar → "File" → "Switch Workspace...".

† If you switched workspace, also terminate external tools `te_vcom` and `gterm`. Then you have to run `te_vcom` and `gterm` again. On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(2) Change system timer interrupt period (TTimPeriod)

Although T-Kernel 2.0 has functions to control time in microseconds, its time resolution depends on the system timer interrupt period (`TTimPeriod`). The default value is 10 milliseconds, namely 10000 microseconds. Here let's change it to 100 microseconds by "`sysconf TTimPeriod` *milliseconds microseconds*" command. The change becomes valid after system reboot.

```
[/SYS]% sysconf TTimPeriod 0 100 ↩
+ 0: TTimPeriod 0 100
[/SYS]% exit ↩
<< EXIT cli >>
[IMS]% exit     1
[IMS]% exit -3 ↩
<< SYSTEM REBOOT >>
```

(3) Create a Project

On the "C/C++ Project" View, click on the mouse right button to show a menu, and select "New" → "T-Engine C/C++ Project". (Or select Menubar → "New" → "T-Engine C/C++ Project".)

Specify the following items.

- Project name :
  Enter the project name, `usec`.

- Target :
  Specify your target machine (pcat).
- Program Type:
  Specify "T-Kernel Base", or "Decide by workspace", if your workspace is `C:\te\kappl`.
- Template:
  Do not check it, because the source program will be expanded later.
- Output Directory:
  Check it.

And click on the Finish button to create a new project.

(4) Expand Source Program

The source program is archived in "`usec.zip`" on the "`en\tutorial`" folder in the T-Kernel 2/x86 Evaluation Kit CD-ROM. Expand this archive "`usec.zip`" to "`C:\te\kappl`" by Windows Explorer.

In the "C/C++ Project" View, select right-click menu → "Refresh". Now the source files appear under the "`src`" directory of the project "`usec`" in the "C/C++ Project" View. Double-click on "`usec`" to view the source files.

(5) Make (compile and link)

Select the target Makefile, `usec/pcat/Makefile`, in the "C/C++ Project" View.

And then, select Menubar → "Project" → "Make T-Engine Target all" to make (compile and link) your program.

If the make is succeeded, the relocatable executable file `usec/pcat/usec` is generated.

(6) Transfer and Execute the Executable File

Select the executable file `usec/pcat/usec`, click on the mouse right button to show a menu, and select "Run As" → "Run Configurations...". Then appears the Run Configurations Dialog.

On the dialog, select "T-Engine Application", click on the mouse right button to show a menu, and select "New". Then appears the default settings to transfer and execute the program.

Click on the Run button at the bottom of the dialog to transfer and execute the program. Then the program is loaded on the system common space.

```
[/SYS]% lodspg usec
SYSPRG usec [14] c0676000 - c067a000
TTimPeriod = 100[us]
#1:  8783629700[us]
#2:  8783629800[us]
#3:  8783629900[us]
#4:  8783630000[us]
#5:  8783630100[us]
[/SYS]%
```

(7) Unload the program

Now unload `usec` by executing the "`unlspg`" command, specifying the program by its system program ID as displayed in the previous step (14 in the above example).

```
[/SYS]% unlspg 14 ↩
[/SYS]%
```

(8) Resume TTimPeriod

Resume the system timer interrupt period (`TTimPeriod`) to the default value, namely 10 milliseconds.

```
[/SYS]% sysconf TTimPeriod 10 ↩
+ 0: TTimPeriod 10
[/SYS]% exit ↩
<< EXIT cli >>
[IMS]% exit     1
[IMS]% exit -3 ↩
<< SYSTEM REBOOT >>
```

4.4 Physical Timer Sample Implementation

This T-Kernel-based sample program uses physical timer to measure time to start task (`tk_sta_tsk`). Note that the resolution of the physical timer does not depend on the system timer interrupt period (`TTimPeriod`).

(1) Select a Workspace

Launch Eclipse, and on the "Select a workspace" dialog, select the workspace "`C:\te\kappl`" for T-Kernel-based programs, and click on OK.

If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\bappl`", by Menubar → "File" → "Switch Workspace...".

† If you switched workspace, also terminate external tools `te_vcom` and `gterm`. Then you have to run `te_vcom` and `gterm` again. On the console view of `gterm`, hit ↩ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(2) Create a Project

On the "C/C++ Project" View, click on the mouse right button to show a menu, and select "New" → "T-Engine C/C++ Project". (Or select Menubar → "New" → "T-Engine C/C++ Project".)

Specify the following items.

- Project name :
  Enter the project name, `ptimer`.
- Target :
  Specify your target machine (pcat).
- Program Type:
  Specify "T-Kernel Base", or "Decide by workspace", if your workspace is `C:\te\kappl`.
- Template:
  Do not check it, because the source program will be expanded later.
- Output Directory:
  Check it.

And click on the Finish button to create a new project.

(3) Expand Source Program

The source program is archived in "`ptimer.zip`" on the "`en\tutorial`" folder in the T-Kernel 2/x86 Evaluation Kit CD-ROM. Expand this archive "`ptimer.zip`" to "`C:\te\kappl`" by Windows Explorer.

In the "C/C++ Project" View, select right-click menu → "Refresh". Now the source files appear under the "`src`" directory of the project "`ptimer`" in the "C/C++ Project" View. Double-click on "`ptimer`" to view the source files.

(4) Make (compile and link)

Select the target Makefile, `ptimer/pcat/Makefile`, in the “C/C++ Project” View.

And then, select Menubar → “Project” → “Make T-Engine Target all” to make (compile and link) your program.

If the make is succeeded, the relocatable executable file `ptimer/pcat/ptimer` is generated.

(5) Transfer and Execute the Executable File

Select the executable file `ptimer/pcat/ptimer`, click on the mouse right button to show a menu, and select “Run As” → “Run Configurations...”. Then appears the Run Configurations Dialog.

On the dialog, select “T-Engine Application”, click on the mouse right button to show a menu, and select “New”. Then appears the default settings to transfer and execute the program.

Click on the Run button at the bottom of the dialog to transfer and execute the program. Then the program is loaded on the system common space.

```
[/SYS]% lodspg ptimer
SYSPRG ptimer [14] c0676000 - c067a000
Physical timer resolution:  167.8[MHz] = 6.0[ns]
GetPhysicalTimerCount:  30.2[count] = 179.7[ns]
GetPhysicalTimerCount + tk_sta_tsk:  142.8[count] = 850.9[ns]
tk_sta_tsk:  112.6[count] = 671.2[ns]
[/SYS]%
```

The above result shows that starting task requires about 670 nanoseconds (0.67 microseconds). Note that getting physical timer count
(`GetPhysicalTimerCount`) also requires time, which must be subtracted.

† This result depends on hardware and software conditions, for example it may be slow on virtual environment (VMware).

(6) Unload the program

Now unload `ptimer` by executing the “`unlspg`” command, specifying the program by its system program ID as displayed in the previous step (14 in the above example).

```
[/SYS]% unlspg 14 ↩
[/SYS]%
```

Index

Getting Started with T-Kernel 2/x86
Version 2.0.1

Web: http://www.t-engine4u.com/
E-Mail: te-sales@personal-media.co.jp