

---

# Getting Started with T-Kernel 2.0 Reference Kit

---



Version 1.01.00



Personal Media Corporation

Copyright © 2011 by Personal Media Corporation

## Contents

<b>History of Changes</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>1 Installing Target</b>	<b>5</b>
1.1 Console Connection . . . . .	6
1.2 Make Boot Disk . . . . .	10
1.3 Font Installation . . . . .	13
<b>2 Installing Development Environment</b>	<b>15</b>
2.1 Installing Cygwin . . . . .	15
2.2 Installing Eclipse . . . . .	17
2.3 Installing T-Kernel Development Environment . . . . .	19
<b>3 Development</b>	<b>24</b>
3.1 Starting Target and Development Environment . . . . .	24
3.2 Console Tools . . . . .	24
3.3 Process-Based and T-Kernel-Based Programs . . . . .	26
3.4 "Hello, world" as a Process-Based Program . . . . .	28
3.5 "Hello, world" as a T-Kernel-Based Program . . . . .	31
<b>4 Running the Learning Sample Programs</b>	<b>34</b>
4.1 Juggling . . . . .	34
4.2 Tiny Web Server . . . . .	37
4.3 LED and Interrupt . . . . .	40
4.4 Physical Timer Sample Implementation . . . . .	42
<b>Index</b>	<b>44</b>

## **History of Changes**

### **Version 1.01.00**

- USB-UART driver is included in T-Kernel 2.0 Reference Kit CD-ROM.
- Add physical timer implementation sample.

### **Version 1.00.00**

- First edition for T-Kernel 2.0 Reference Kit.

## **Introduction**

This document provides a tutorial on starting out with T-Kernel 2.0 Reference Kit, for those using this product for the first time.

For development environment, both Windows and Linux are supported. In this document, however, we introduce a tutorial only on Windows with Eclipse.

## 1 Installing Target

In general on developing embedded systems, the development environment is apart from the execution target. This style is called “cross-development”, shown in Figure 1.1. In case of T-Kernel 2.0 Reference Kit, we also use this style for software development.

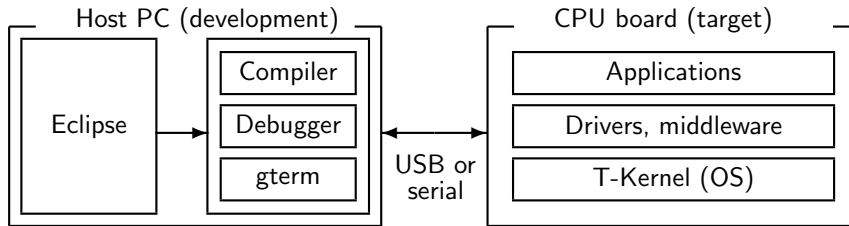


Figure 1.1 Cross-development

In this section we’ll install the target-side environment (OS, drivers and middleware) on a microSD for Reference Board, while the section 2 describes the host-side installation. For details please also refer to “*T-Kernel 2.0 Reference Kit Manual*” at “1. Introduction”.

Parts on the Reference Board are shown in Figure 1.2.

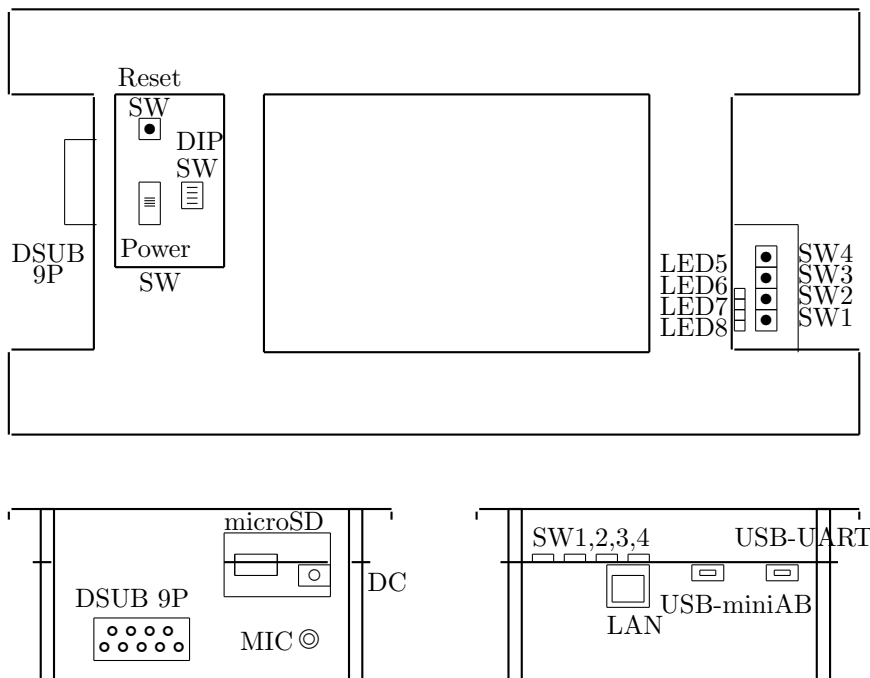


Figure 1.2 T-Kernel 2.0 Reference Kit CPU board

## 1.1 Console Connection

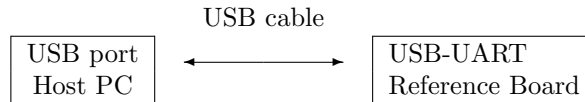
Both USB and serial (RS-232C) connections are supported for console of Reference Board.

### (1) Connection

#### ○ In case of USB connection

For the first time it is necessary to install device driver to Windows.

1. Log on Windows as an administrator. “common\soft” folder of the T-Kernel 2.0 Reference Kit CD-ROM contains the USB-UART (CP210x) driver installer, “CP210x\_VCP\_Win\_XP\_S2K3\_Vista\_7.exe”. Run this executable file and install the driver.
  - \* If you prefer the newest one, it will be get from the URL below at Silicon Laboratories Inc.:  
<http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>
2. Run the downloaded executable file and install it.
3. Supply power to DC jack of Reference Board, and slide POWER SW to the upper position. Connect USB port of your host PC and USB-UART connector of Reference Board(Figure 1.3).



**Figure 1.3 USB connection**

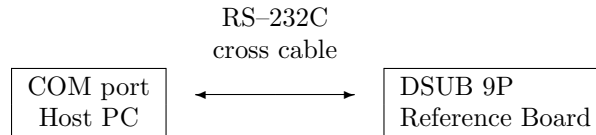
4. Windows detects Reference Board as a USB device, and installs device driver.
5. Open “Device Manager” (in case of Windows 7 and Vista: “Control Panel” → “System and Security” → “Device Manager”, in case of Windows XP: “Control Panel” → “System and Security” → “System” → “Hardware” → “Device Manager”). See “Port (COM and LPT)” and check that “Silicon Labs CP210x USB to UART Bridge (COM $n$ )” is installed. Here COM $n$  is the COM port number.

### Remarks on USB Connection

- It is recommended to connect PC and Reference Board directly by USB cable (without USB hub). USB hub may cause some problems.
- Boot/reboot/reset must be done while pushing SW2 to connect by USB. Without pushing SW2 serial connection is selected by default setting. But if you write ROM information for USB connection to flash ROM, you can connect by USB without pushing SW2. See page 9.
- If you boot/reboot/reset Reference Board, USB connection may be disconnected. Then quit and restart your terminal software.

○ **In case of serial connection (RS-232C)**

Connect COM port of your host PC and DSUB 9P connector of Reference Board by serial (RS-232C) cross cable (Figure 1.4).



**Figure 1.4 Serial connection**

**(2) Boot Reference Board**

Supply power to DC jack of Reference Board, and slide POWER SW to the upper position.

In case of USB connection, start supplying power while pushing SW2.

**(3) Start Terminal Software**

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

† In cases of Windows 7 and Windows 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>

† If you install Eclipse Environment, you can use a terminal software “gterm” on Eclipse. But for the first setup, we recommend to use a simple terminal software (such as Hyper Terminal) for easy trouble shooting.

● **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 . (Figure 1.5).

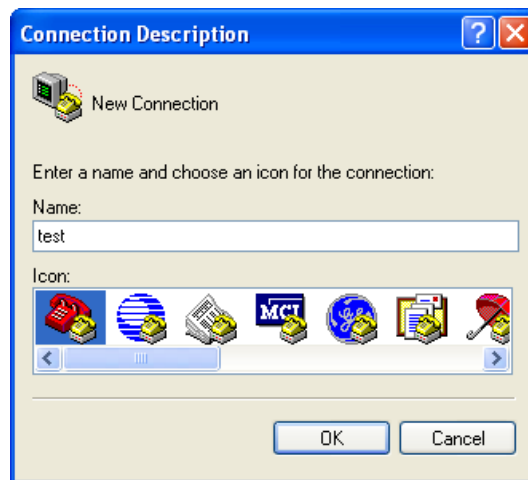


Figure 1.5 Start HyperTerminal

#### (4) Communication Setting

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	XON/XOFF

† Hardware flow control (RTS/CTS) is not supported. Please use XON/XOFF control instead.

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

Select “File” → “New connection”. On the new connection dialog, check “Serial”, and select the serial port (ex. COM1), and click . Then select “Setup” → “Serial port”, and set parameters as above.

† If you use Tera Term Pro and the COM port number is above the range that can be specified, change the `MaxComPort` setting in the configuration file `TERATERM.INI` (normally located in `C:\Program Files\TTERMPRO`), using a text editor (Notepad, etc.).

- **In case of HyperTerminal :**

Select the serial port (ex. COM1) and click . Then set the communication parameters (Figure 1.5).



Figure 1.6 Communication Setting

**(5) Confirm Connection**

Try pressing  $\leftrightarrow$  (Enter) Key a number of times on the terminal software. The CLI prompt ([/SYS]%) will appear on the terminal software (Figure 1.7).

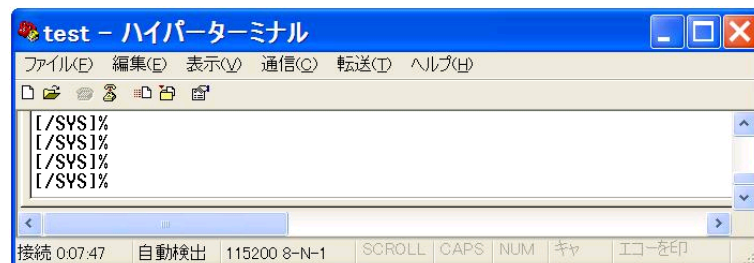


Figure 1.7 CLI prompt on a terminal software

**(6) Write ROM Information (only for USB connection)**

The “en\soft” folder of the T-Kernel 2.0 Reference Kit CD-ROM contains the following files to write ROM information on flash ROM for switching USB connection and serial connection.

rominfo.mot	serial	(USB: boot with SW2)	default
rominfo-usb.mot	USB	(serial: boot with SW2)	

Without pushing SW2 serial connection is selected by default setting. But if you write ROM information for USB connection to flash ROM, you can connect by USB without pushing SW2.

Push reset switch while pushing both SW1 and SW2 to start T-Monitor. Try pressing  $\leftrightarrow$  (Enter) Key a number of times on the terminal software. The T-Monitor prompt (TM>) will appear.

Input FlashLoad command on T-Monitor.

```
TM> FlashLoad↵
Copy Flash ROM Image to RAM Area
> Load S-Format Data of Flash ROM
```

After a moment a message “> Load S-Format Data of Flash ROM” will appear. Then send the file rominfo-usb.mot in T-Kernel 2.0 Reference Kit CD-ROM.

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

Select Menubar → “File” → “Send file...”, and specify the file to be sent (rominfo-usb.mot).

- **In case of HyperTerminal :**

Select Menubar → “Transfer” → “Send Text File”, and specify the file to be sent (rominfo-usb.mot). Select \*.\* for file name suffix, because it is not .txt.

```
Loaded: 70020000 -> 7002007F
Writing Flash ROM at 70020000 [1 blks] ... wait
TM>
```

After writing flash ROM appears T-Monitor prompt (TM>). From now on you can connect by USB without pushing SW2.

## 1.2 Make Boot Disk

### (1) Insert microSD Card

Insert a blank (to be erased) microSD card to Reference Board.

† eMMC and USB storage devices are also available instead of microSD. For more details refer to “T-Kernel 2.0 Reference Kit Manual” at “1.5 Making Boot Disk”.

### (2) Make Partition

Run `hdpart` command from the terminal software, where the device name is “pcb”, which means the whole disk.

```
[/SYS]% hdpart pcb↵ — run hdpart command
pcb [C:485 H:64 S:63 B:1961984 (958 MB)]
No System Boot StartCHS EndCHS SecNo SecCnt Size
1 0b DOS 00 0: 1: 1 486: 38:38 63 1961921 957 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/Quit ? d↵ — delete partition
Delete PartNo (1-4) ? 1↵ — 1st partition
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] (<957MB) ? a↵ — partition size
No System Boot StartCHS EndCHS SecNo SecCnt Size
1 13 BTRON 00 0: 1: 1 486: 38:38 63 1961921 957 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 partition
Boot PartNo (1-4,Clear) ? 1↵ — 1st partition
No System Boot StartCHS EndCHS SecNo SecCnt Size
1 13 BTRON 80 0: 1: 1 486: 38:38 63 1961921 957 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
** pcb: Updated Master Boot Block
[/SYS]%

```

**(3) Format**

Format the 1st partition by `format` command from the terminal software. Device name is “pcb0”. And “-b” option (write boot record) is required. Also “-x” option is required if the partition size is 2GB or more.

```

[/SYS]% format -b pcb0 SYSTEM↵ — format 1st partition
Format pcb0 [STD] SYSTEM
Logical Formatting...
Writing BootCode...
Disk Format Success.
[/SYS]%

```

**(4) Copy the ROM Disk Files**

Copy the all ROM disk files to the 1st partition.

```

[/SYS]% att pcb0 A↵ — attach 1st partition
pcb0 -> A
[/SYS]% rcp -r /SYS /A/=↵ — copy ROM disk files
[/SYS]% det pcb0↵ — detach 1st partition
[/SYS]%

```

**(5) Reboot Reference Board**

Use “exit” and “exit -1” commands to reboot Reference Board.

```

[/SYS]% exit↵
<< EXIT cli >>
[IMS]% exit 1
[IMS]% exit -1↵
<< SYSTEM RESTART >>
:
[/SYS]%

```

The terminal software of the host PC shows booting messages of T-Kernel, and the CLI prompt ([/SYS]).

**(6) Confirm Booting from microSD**

Use the CLI command “df” to confirm that the system (/SYS) was booted from the 1st partition on the microSD card (logical device name is pcb0).

If the device name is rda, it means ROM disk. The system failed to boot from microSD card. Please check the whole procedure above again.

```
[/SYS]% df↵
PATH DEV      TOTAL      FREE USED   UNIT MAXFILE NAME
/SYS pcb0    980960    977108  0% 4096  65535 SYSTEM
[/SYS]%
```

**(7) Copy T-Shell Install Package Files**

Prepare a FAT-formatted USB storage (USB memory etc.). Copy the following 2 files from the “en\soft” folder of the T-Kernel 2.0 Reference Kit CD-ROM to a USB storage on PC.

File name	Content
install_tshell.bz	T-Shell install package
english.bz	T-Shell English patch

† Instead of file copy on USB storage, serial/USB transfer in XMODEM protocol and LAN transfer in FTP are also available. For more details refer to “*T-Kernel 2.0 Reference Kit Manual*” at “1.5.2 Installing T-Shell”.

**(8) Install T-Shell**

Connect this USB storage to the USB miniAB connector of Reference Board. (Use a cable to connect between your USB storage and USB miniAB connectors.)

```
[/SYS]% att -m uda0 uda0↵      — attach USB storage (FAT)
uda0 -> uda0
[/SYS]% expf -U /uda0/install_tshell.bz↵ — expand package
[/SYS]% install_tshell /SYS↵      — install T-Shell
Install completed, please reboot the system.
[/SYS]% rm -r install_tshell↵      — remove T-Shell installer
[/SYS]%
```

Now use English patch before rebooting the system. This English patch translates T-Shell GUI applications to English.

```
[/SYS]% expf -U /uda0/english.bz↵ — expand English patch
[/SYS]% english /SYS↵      — install English patch
.. Installing English Patch on /SYS
.. Install completed
[/SYS]% rm -r english↵      — remove English patch
[/SYS]% det -u uda0↵      — detach USB storage
[/SYS]%
```

The USB storage can be disconnected after det command.

**(9) Confirm T-Shell**

Use “exit” and “exit -1” commands to reboot Reference Board.

```

[/SYS]% exit↵
<< EXIT cli >>
[IMS]% exit 1
[IMS]% exit -1↵
<< SYSTEM RESTART >>
:
[/SYS]%

```

The terminal software of the host PC shows booting messages of T-Kernel, and the CLI prompt ([/SYS]%).

And the LCD screen of Reference Board shows an initial window of PMC T-Shell (Figure 1.8).

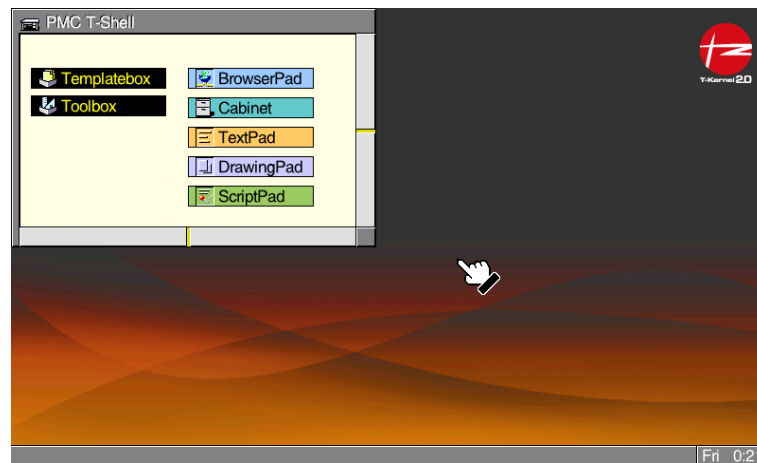


Figure 1.8 Initial Window of PMC T-Shell

Touch the system message panel at the bottom of the screen to show menu. Select menu → “Utility” → “Touch Panel Adj.” to adjust touch panel coordination. This adjustment becomes valid after system reboot.

### 1.3 Font Installation

The “common\font” folder of the T-Kernel 2.0 Reference 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
SSHVELVET	SSHelvetica
SSHVELVTN	SSHelveticaN
SSTIMES	SSTimes

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

**(1) Copy Font File**

Copy the following 2 files from the “en\soft” folder of the T-Kernel 2.0 Reference Kit CD-ROM to a USB storage on PC.

File name	Content
instfont	font installer
SSHVELVET	SSHelvetica font

**(2) Install Font**

Connect this USB storage to the USB miniAB connector of Reference Board. (Use a cable to connect between your USB storage and USB miniAB connectors.)

```
[/SYS]% att -m uda0 uda0↵          — attach USB storage (FAT)
uda0 -> uda0
[/SYS]% /uda0/instfont /uda0/SSHVELVET↵          — install font
Font File Name      : regist
0: **-****          : use
1: ****-SSHelvetica** : use
[/SYS]% det -u uda0↵          — detach USB storage
[/SYS]%
```

**(3) Reboot Reference Board**

Use “exit” and “exit -1” commands to reboot Reference Board.

```
[/SYS]% exit↵
<< EXIT cli >>
[IMS]% exit 1
[IMS]% exit -1↵
<< SYSTEM RESTART >>
:
[/SYS]%
```

## 2 Installing Development Environment

### 2.1 Installing Cygwin

Cygwin is required for the Reference Board Development Environment on PC. For more details please also refer to the “*Cygwin Install Manual*” in the T-Kernel 2.0 Reference Kit CD-ROM.

† Windows Logon 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 2.0 Reference Kit CD-ROM.

Cygwin System Package	common\soft\cygwin\_d.1.7.7-1.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)

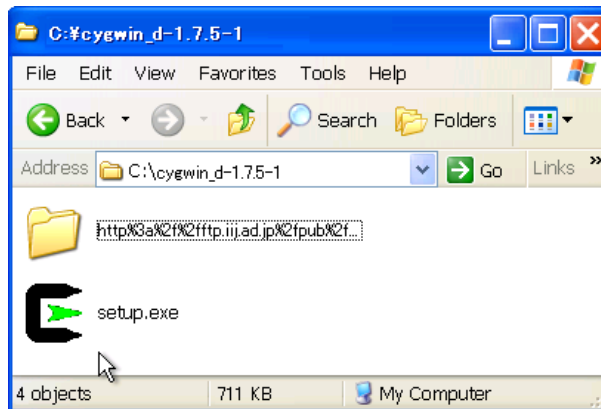


Figure 2.1 Start Cygwin Installer

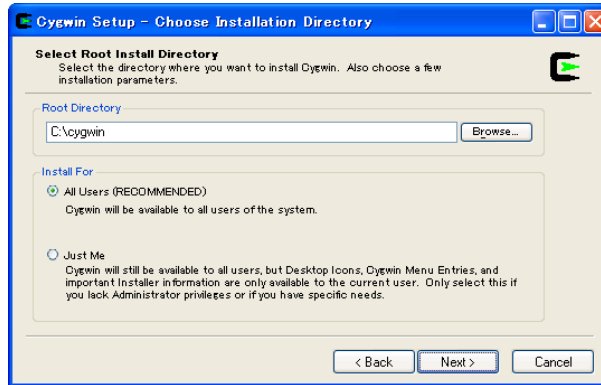
Click on `Next >`.

#### (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. It can be left as is. (Figure 2.2)



**Figure 2.2** Select Install Directory

Select All Users at “Install For” (default choice). And click on .

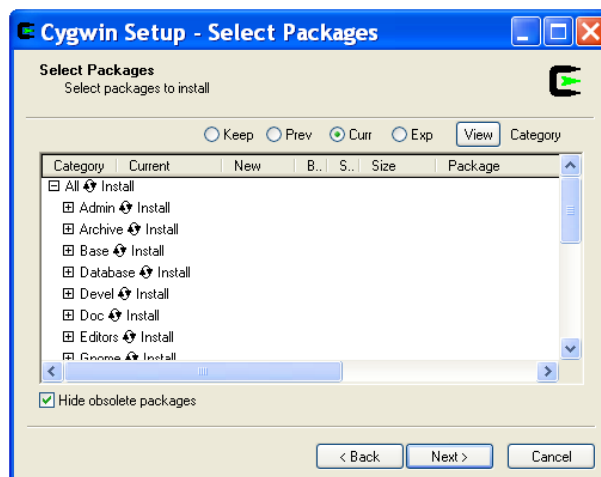
**(5) Select Local Package Directory**

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

Click on  to continue.

**(6) Select Install Packages**

After checking files, a “Setup Alert” dialog may appear. If there is no problem, click on . Next appears a screen for selecting packages to install. The initial selected state is “All  Default”. Click on “Default” to change to “Install”. (Figure 2.3)



**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`.

**(9) Distinguish Uppercase and Lowercase Filenames**

Cygwin does not distinguish uppercase and lowercase filenames.

To distinguish them, open the startup batch file `C:\cygwin\Cygwin.bat` by a text editor, and before the line

```
bash --login -i n
add
set CYGWIN=nowinsymlinks check_case:strict
.
```

**(10) Initialize Home Directory**

After modifying the batch file, 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.

**(11) Create a symbolic link for gmake**

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

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

**(12) 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.

```
$ exit↵
```

**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 2.0 Reference 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 following website to download Java Runtime.

<http://www.java.com/en/>

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

## (2) Expand Eclipse System Package

Copy (save) the following files to a temporary directory of your install drive (such as C:\temp) from the common software directory (common\soft) of the T-Kernel 2.0 Reference Kit CD-ROM.

Eclipse	common\soft\eclipse-platform-3.2.2-win32.zip
CDT plug-in	common\soft\org.eclipse.cdt-3.1.2-win32.x86.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.

## (4) Confirming for running Eclipse

Double-click on the shortcut of Eclipse to launch Eclipse (Figure 2.4). On the “Select a workspace” dialog, click on **Cancel** to quit Eclipse.



Figure 2.4 Starting Eclipse

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

## 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 2.0 Reference 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 2.0 Reference Kit CD-ROM.

Common part for Eclipse platform	<code>com.t_engine4u.te.1.1.0.zip</code>
Machine-dependent part	<code>com.t_engine4u.tl.em1d512_te.1.0.1.zip</code>

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 2.0 Reference Kit CD-ROM.

T-Shell additional package	<code>com.t_engine4u.tsh.em1d512_te.1.0.0.zip</code>
----------------------------	--

And expand this archive to  
`C:\eclipse\plugins\com.t_engine4u.tl.em1d512_te.1.0.1_1.0.1\te`  
 by Windows Explorer.

† Version numbers may be different.

### (3) Start Eclipse

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

### (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 .

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

<code>C:\te\bappl</code>	Process-based programs
<code>C:\te\kappl</code>	T-Kernel-based programs
<code>C:\te\driver</code>	Device drivers

**(5) Go to the Workbench**

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

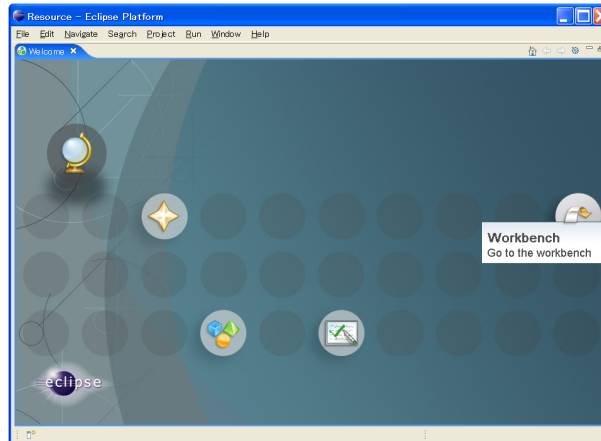


Figure 2.5 “Welcome” view of Eclipse

**(6) Open the “T-Engine Development” Perspective**

Select Menubar → “Window” → “Open Perspective” → “T-Engine Development”. (Figure 2.6)

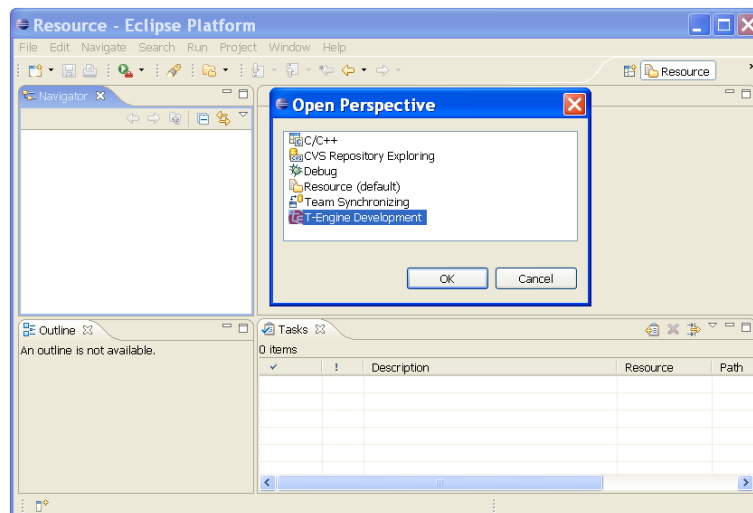
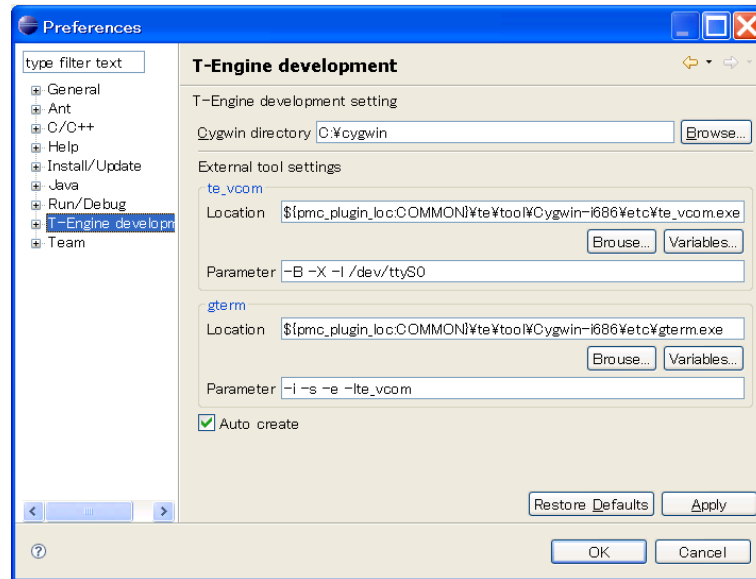


Figure 2.6 Select T-Engine development perspective

**(7) Setting of T-Engine Development**

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



**Figure 2.7 T-Engine Development Settings**

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

† “`-B`” means 115200 bps, “`-X`” means suppressing hardware flow control (RTS/CTS), “`-l`” is minus ell, which stands for line device, and “`/dev/ttySn`” means COM( $n + 1$ ).

#### (8) Setting of Workspace

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

- Click on “Build automatically” to be OFF.
- Select “New text file line delimiter” to be Other → Unix.

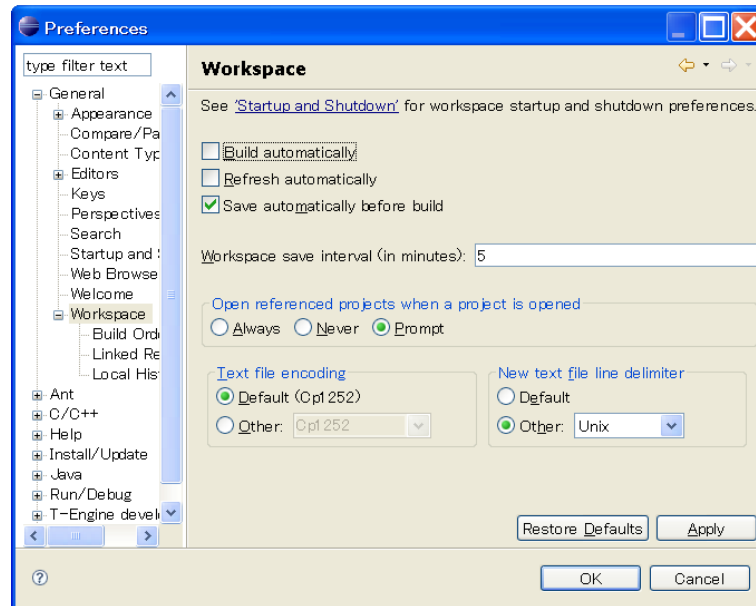


Figure 2.8 Setting of Workspace

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

#### (9) Opening console

Click on the “External Tools” button of Toolbar, and select “External Tools...” in the pull-down menu. (Figure 2.9)

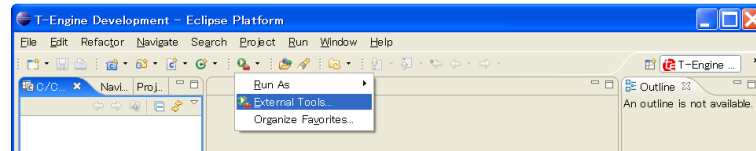


Figure 2.9 External Tools

And double-click on the “Program”. Then appear te\_vcom and gterm under “Program”.

Select te\_vcom and click on the **Run** button. (Figure 2.10)

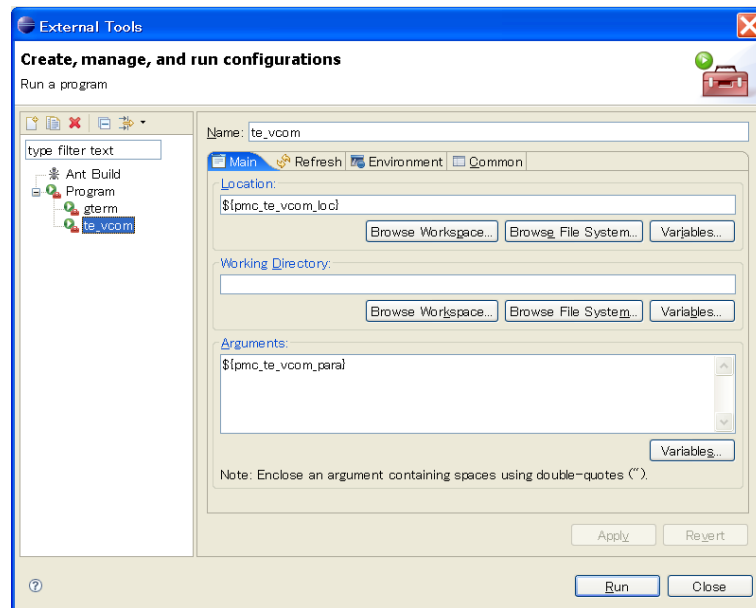


Figure 2.10 Running te\_vcom

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

On the console window of gterm, hit ↵ (Enter) key several times. Then appears a prompt (such as [/SYS]%) from the target side (Reference Board). This is a prompt of CLI (Command Line Interpreter), which runs on Reference Board. (Figure 2.11)

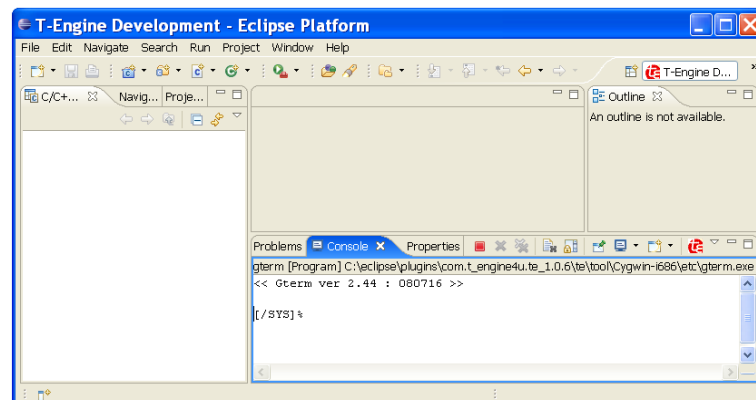


Figure 2.11 CLI prompt on gterm

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 Reference Board and Eclipse as follows.

#### (1) Boot the target side (Reference Board)

Supply power to DC jack of Reference Board, and slide POWER SW to the upper position.

The LCD screen of Reference Board shows an initial window of 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 window of `gterm`, hit `↵` (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`) from the target side (Reference Board).

† 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 by the reset switch of Reference Board.

### 3.2 Console Tools

T-Kernel 2.0 Reference Kit 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 2.0 Reference Kit 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 window 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) T-Monitor Command Help**

Use the T-Monitor command “?” to display details of the “iw” command.

```
TM> ? iw↵
InputWord(IW) port : Input Word from I/O port
```

**(8) I/O read**

Use the T-Monitor command “iw” to read 32 bit value from address 0xc0050010 (GIO[31:0] input data read register) to read status of push switches SW1,2,3,4.

```
TM> iw c0050010↵  — SW1 pushed  SW1  bit8  0x00000100
Port C0050010:W --> 00000101      SW2  bit7  0x00000080
TM> iw c0050010↵  — SW2 pushed  SW3  bit6  0x00000040
Port C0050010:W --> 00000081      SW4  bit4  0x00000010
```

**(9) 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 Reference Board are generally classified as either process-based programs or T-Kernel-based programs.

**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 2.0 Reference 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 2.0 Reference 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 .

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 window 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. (Figure 3.1)

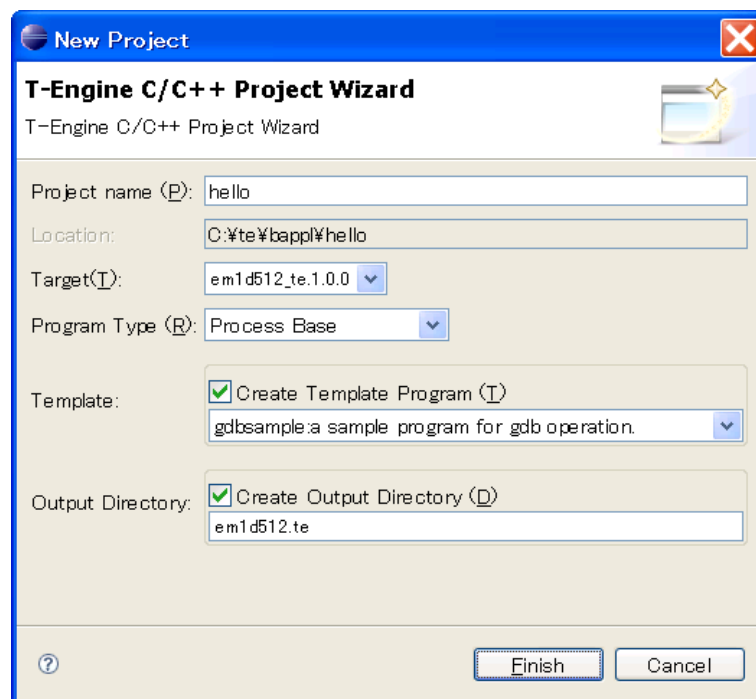


Figure 3.1 New Project Dialog

- Project name :  
Specify your project name, for example, `hello`.

- Target :  
Specify your target machine (em1d512.te).
- 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 type “em1d512.te”.

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 Menubar → “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 Menubar → “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/em1d512.te/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 executable file `hello/em1d512.te/hello` is generated.

† Process executable file is shown with a runner or bug icon. The “bug icon” means that some of linked libraries include debug information, but it is all right.

**[Trouble Shooting]** If appears a message “Error launching builder”, confirm that you have selected the make target `hello/em1d512.te/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/em1d512.te/hello`, click on the mouse right button to show a menu, and select “Run As” → “Run...”. Then appears the Run Dialog.

On the Run 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)

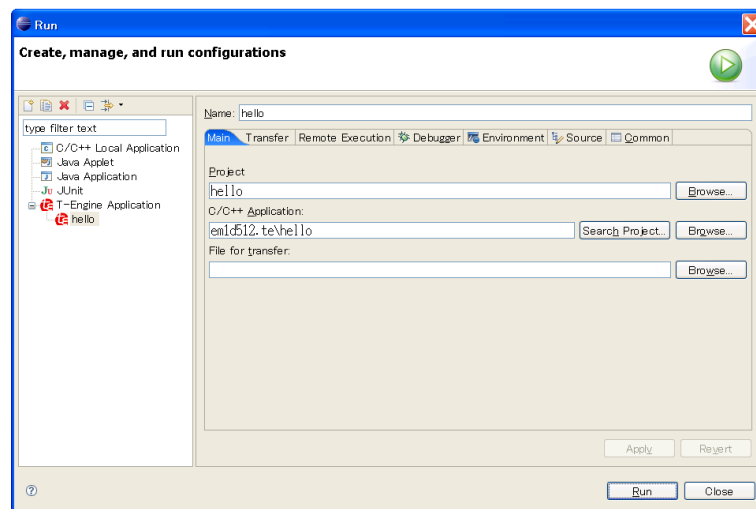


Figure 3.2 Run Dialog

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

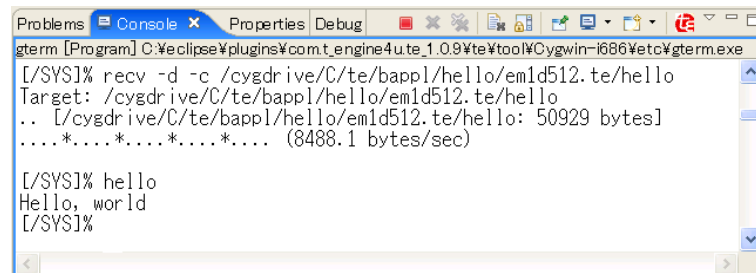


Figure 3.3 “Hello, world” appears on console

**[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 window 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 Reference Board 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\kapp1`” for T-Kernel-based programs, and click on .

If you have already launched Eclipse and works on other workspace, switch workspace to “`C:\te\kapp1`”, 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 window 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 :  
Specify your project name, for example, `hello2`.
- Target :  
Specify your target machine (`em1d512.te`).
- Program Type:  
Specify “T-Kernel Base”, or “Decide by workspace”, if your workspace is `C:\te\kapp1`.
- Template:  
Check it and select the template “sample: print strings sample program”.
- Output Directory:  
Check it, and type “`em1d512.te`”.

And click on the  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 Menubar → “File” → “Save”.

- `hello2/src/Makefile` :

Change the make target to “TARGET = `hello2`”.

Save the modified file by Menubar → “File” → “Save”.

**(4) Make (compile and link)**

Select the target Makefile, `hello2/em1d512.te/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 `hello2/em1d512.te/hello2` is generated. And `hello2.abs` is a absolute address executable file, and `hello2.trg` is its stripped file.

† Eclipse shows T-Kernel-based relocatable executable files by a “binary file icon” (not runner or bug icon).

**(5) Transfer and Execute the Executable File**

Select the executable file `hello2/em1d512.te/hello2`, click on the mouse right button to show a menu, and select “Run As” → “Run...”. Then appears the Run Dialog.

On the Run 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 Run 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] - 0xd018b000 - 6  screen  
... omitted ...  
[13] - 0xd049d000 - 5  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 (13 in the above example). Then appears “Hello, world” on the gterm console.

```
[/SYS]% unlspg 13  
See you again  
[/SYS]%
```

## 4 Running the Learning Sample Programs

The tutorial page (`en\tutorial`) on the T-Kernel 2.0 Reference Kit CD-ROM contains a collection of learning sample programs.

- **Juggling**

This program shows juggling animation.

- **Tiny Web Server**

This program transfers files from Reference Board to Web browsers on other PCs by TCP/IP network.

- **LED and Interrupt**

This T-Kernel-based program blinks LED and detects switch by interrupt.

- **Physical timer sample implementation**

This T-Kernel-based program uses 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 .

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 window 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, `juggling`.
- Target :  
Specify your target machine (`em1d512.te`).
- 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 type “em1d512.te”.

And click on the  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 2.0 Reference 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 → “Update”. 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/em1d512.te/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 executable file “juggling/em1d512.te/juggling” is generated (with a runner or bug icon).

### (5) Transfer and Execute the Executable File

Select the executable file “juggling/em1d512.te/juggling”, click on the mouse right button to show a menu, and select “Run As” → “Run...”. Then appears the Run Dialog.

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

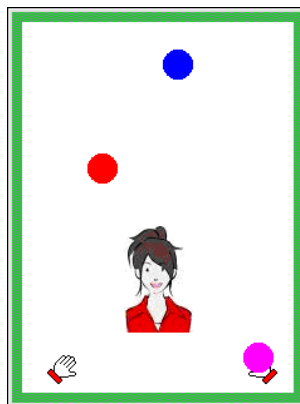


Figure 4.1 Run Juggling

**(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 “em1d512.te.debug”.

Select the target Makefile, “juggling/em1d512.te.debug/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 executable file “juggling/em1d512.te.debug/juggling” is generated with debug information (showed by a insect icon).

**(7) Start Debugging**

Select the executable file “juggling/em1d512.te.debug/juggling”, click on the mouse right button to show a menu, and select “Debug As” → “Debug...”. Then appears the Debug Dialog.

On the Debug 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.

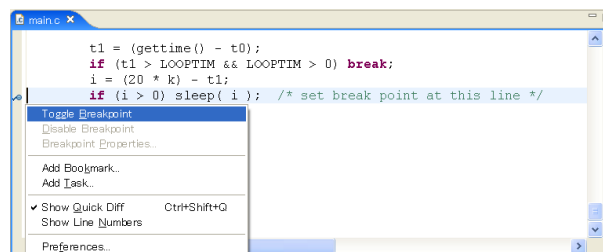
† There remains a previous setting created in (5) for execution. But create a new one for debugging.

Click on the **Debug** button at the bottom of the Debug 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.

**(8) Set a Breakpoint**

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

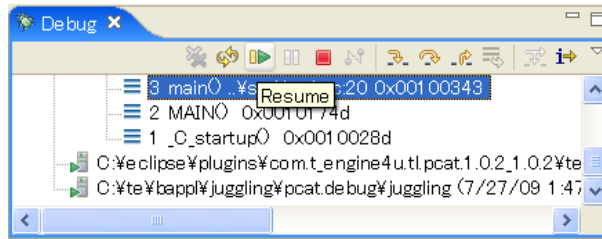
Now set a breakpoint in the line “`if (i > 0) sleep( i );`” near the end of the `main()` function. At the leftmost side of this line, click on the mouse right button to show a menu. Select “Toggle Breakpoint” to set a breakpoint. (Figure 4.2)



**Figure 4.2 Set a Breakpoint**

**(9) 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)

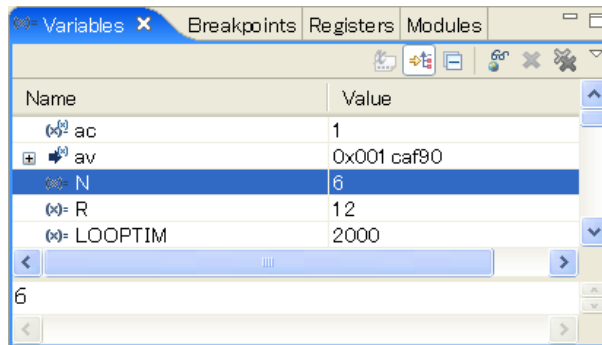


**Figure 4.3 Continue Execution**

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

#### (10) 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)



**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).

#### (11) 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 Reference Board 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\bapp1” for process-based programs, and click on **OK**.

If you have already launched Eclipse and works on other workspace, switch workspace to “C:\te\bapp1”, 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 window of `gterm`, hit ↵ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

**(2) Network Configuration**

On the console window 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 Reference Board.

```
[/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 window 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 ↵ — ping another machine
192.168.0.1 is alive <192.168.0.1> : 0 ms
```

† In this example, the Reference Board 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.

**(4) Transfer Image Files**

Transfer Image Files to the target side (Reference Board), for the contents of the tiny web server.

On the console window of gterm, use the “`recv`” command to transfer image files from the host PC to the target side (Reference Board).

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 Menubar → “New” → “T-Engine C/C++ Project”.)

Specify the following items.

- Project name :  
Enter the project name, `webserv`.
- Target :  
Specify your target machine (`em1d512.te`).
- 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 type “`em1d512.te`”.

And click on the  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 2.0 Reference 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 → “Update”. 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/em1d512.te/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 executable file “`webserv/em1d512.te/webserv`” is generated (with a runner or bug icon).

**(8) Transfer and Execute the Executable File**

Select the executable file “webserv/em1d512.te/webserv”, click on the mouse right button to show a menu, and select “Run As” → “Run...”. Then appears the Run Dialog.

On the Run 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 Run Dialog to transfer and execute the program.

**(9) View from Web Browser**

From another PC on the network, enter the Reference Board IP address in a Web browser, and confirm that the images loaded earlier are listed in the browser. If, for example, the Reference Board 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)

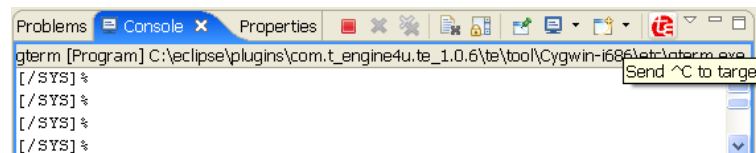


Figure 4.5 Send `^C` on Eclipse

**4.3 LED and Interrupt**

This is a T-Kernel-based sample program to blink LED and detect switch by interrupt.

**(1) Select a Workspace**

Launch Eclipse, and on the “Select a workspace” dialog, select the workspace “C:\te\kapp1” 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\bapp1”, 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 window 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, `ledint`.
- Target :  
Specify your target machine (`em1d512.te`).
- 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 type “`em1d512.te`”.

And click on the  button to create a new project.

**(3) Expand Source Program**

The source program is archived in “`ledint.zip`” on the “`en\tutorial`” folder in the T-Kernel 2.0 Reference Kit CD-ROM. Expand this archive “`ledint.zip`” to “`C:\te\kappl`” by Windows Explorer.

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

**(4) Make (compile and link)**

Select the target Makefile, `ledint/em1d512.te/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 `ledint/em1d512.te/ledint` is generated.

**(5) Transfer and Execute the Executable File**

Select the executable file `ledint/em1d512.te/ledint`, click on the mouse right button to show a menu, and select “Run As” → “Run...”. Then appears the Run Dialog.

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

By this program LED5, LED6, LED7 or LED8 starts blinking. By pushing SW4 changes blinking pattern.

#### 4.4 Physical Timer Sample Implementation

This is a T-Kernel-based sample program for physical timer.

There are several methods and implementations for physical timer, depending on hardware and purpose. Hence T-Kernel 2.0 Reference Kit 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.

##### (1) Select a Workspace

Launch Eclipse, and on the “Select a workspace” dialog, select the workspace “C:\te\kapp1” for T-Kernel-based programs, and click on .

If you have already launched Eclipse and works on other workspace, switch workspace to “C:\te\bapp1”, 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 window 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 (`em1d512.te`).
- Program Type:  
Specify “T-Kernel Base”, or “Decide by workspace”, if your workspace is `C:\te\kapp1`.
- Template:  
Do not check it, because the source program will be expanded later.
- Output Directory:  
Check it, and type “`em1d512.te`”.

And click on the  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 2.0 Reference Kit CD-ROM. Expand this archive “ptimer.zip” to “C:\te\kapp1” by Windows Explorer.

In the “C/C++ Project” View, select right-click menu → “Update”. 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/em1d512.te/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/em1d512.te/ptimer is generated.

**(5) Transfer and Execute the Executable File**

Select the executable file ptimer/em1d512.te/ptimer, click on the mouse right button to show a menu, and select “Run As” → “Run...”. Then appears the Run Dialog.

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

```
[/SYS]% lodspg ptimer
Physical timer number=1, clock=28672000[Hz], limit=2866
SYSPRG ptimer [14] d04a1000 - d04a7000
[/SYS]% Hello 10000
Hello 20000
Hello 30000
:
```

Physical timer handler is called in every 100 microseconds, and printf is called in every 10000th (1 second) time.

# Index

	<b>B</b>		physical timer .....	42
breakpoint .....		36	<b>ping</b> command .....	38
	<b>C</b>		process-based .....	26
CLI .....		24	project .....	28
console .....		24		<b>R</b>
continue .....		36	<b>recv</b> command .....	39
cross-development .....		5	reset switch .....	5, 24
Cygwin .....		15	ROM information .....	9
	<b>D</b>			<b>T</b>
debug information .....		36	TCP/IP .....	37
df command .....		12	Tera Term Pro .....	7
	<b>E</b>		te_vcom .....	24
Eclipse .....		17	T-Kernel-based .....	27
	<b>F</b>		T-Monitor .....	25
FlashLoad command .....		10		<b>U</b>
font .....		13	<b>unlspg</b> command .....	27
format command .....		11		<b>W</b>
	<b>G</b>		web server .....	37
gterm .....		24	workspace .....	19
	<b>H</b>			
hdpart command .....		10		
HyperTerminal .....		7		
	<b>I</b>			
IMS .....		25		
	<b>J</b>			
Java Runtime .....		18		
juggling .....		34		
	<b>L</b>			
LED .....		5, 40		
lodspg command .....		27		
ls command .....		25		
	<b>M</b>			
memory protection .....		26		
microSD .....		10		
	<b>N</b>			
netconf command .....		38		
	<b>P</b>			
Perl .....		17		

---

Getting Started with T-Kernel 2.0 Reference Kit  
Version 1.01.00

Web: <http://www.t-engine4u.com/>  
E-Mail: [te-sales@personal-media.co.jp](mailto:te-sales@personal-media.co.jp)

Copyright © 2011 by Personal Media Corporation

---