

Getting Started with Teaboard (on Eclipse)

Version 1.A0.05

Contents

History of Changes	3
Introduction	4
1 Setup	5
1.1 Connecting Teaboard with PC by USB	5
1.2 Installing Cygwin	9
1.3 Install Eclipse and Teaboard Development Environment	12
1.4 Initial Configuration	14
1.5 How to Start Next Time	17
1.6 Boot from SD Card	18
2 Development	21
2.1 Console Tools	21
2.2 Process-Based and T-Kernel-Based Programs	23
2.3 "Hello, world" as a Process-Based Program	24
2.4 "Hello, world" as a T-Kernel-Based Program	27
2.5 Running the Learning Programs	30
2.5.1 Roulette	30
2.5.2 Timer	31
2.5.3 Tiny Web Server	34
Index	38

History of Changes

Version 1.A0.05

- Describe Windows Vista and Windows 7.

Version 1.A0.04

- Describe Windows Vista.

Version 1.A0.03

- If you connect Teaboard with host PC by USB, direct connection (without USB hub) is recommended.
- If you connect Teaboard with host PC by USB, the connection is kept even on resetting Teaboard of the latest version software. You do not have to terminate and restart terminal software `te_vcom` and `gterm`.

Version 1.A0.03

- If you connect Teaboard with host PC by USB, direct connection (without USB hub) is recommended.
- If you connect Teaboard with host PC by USB, the connection is kept even on resetting Teaboard of the latest version software. You do not have to terminate and restart terminal software `te_vcom` and `gterm`.

Version 1.A0.02

- On creating new partition by `hdpart` command, one have to delete the current partition before creating new one.

Version 1.A0.01

- Install Cygwin and Eclipse from CD-ROM.
- Change archive format of the exercise programs to zip.

Version 1.A0.00

- First edition for Eclipse Environment.

Introduction

This document provides a tutorial on starting out with Teaboard, for those using it for the first time.

† The development environment for Teaboard works on Windows XP SP3, Windows Vista and Windows 7.

This tutorial is outlined below:

- Setup

1. Connecting Teaboard with PC by USB
Install a device driver on Windows to connect Teaboard with PC by USB.
2. Install Cygwin
Cygwin is required for the Teaboard Development Environment on PC.
3. Install Eclipse and Teaboard Development Environment
Install Eclipse and add plug-ins for Teaboard Development, and setup the environment.
4. Boot from SD Card
Make a working disk on SD Card.

- Development

1. Open console
Use a console on Eclipse to communicate with Teaboard.
2. “Hello, world” as a Process-Based Program
Create a new process-based program on Eclipse, and transfer and execute it on Teaboard.
3. “Hello, world” as a T-Kernel-Based Program
The same case but as a T-Kernel-based program.
4. Running the Learning Programs
Make and execute the learning programs (Roulette, Timer, and Tiny Web-server).

1 Setup

1.1 Connecting Teaboard with PC by USB

Install a device driver on Windows to connect Teaboard with PC by USB.

† Both USB and serial (RS-232) can be used to connect PC with Teaboard. But this tutorial uses USB.

Let's start by connecting Teaboard to a PC running Windows. For more details please also refer to the steps described in the *“Teaboard Manual”* at 1.2.1, “In case of USB connection to Windows host”.

† If you connect Teaboard with host PC by USB, direct connection (without USB hub) is recommended, because USB hub may cause some problems.

(1) Install Driver

In case of Windows XP SP3

1. Start “Found New Hardware Wizard”

On the Teaboard DIP switch DSW2, set both positions 1 and 2 to ON. Connect Teaboard and the Windows PC with a USB cable.

Then Windows detects Teaboard, and starts the “Found New Hardware Wizard”. (Figure 1.1)

Does not connect to Windows Update, and click on **Next >**.



Figure 1.1 Found New Hardware Wizard

2. Select Driver Setup File

Select “Install from a list or specific location (Advanced)” and click on **Next >**.

Select “Include this location in the search,” navigate to the folder containing the driver setup file `teaboard.inf`, then click on **Next >**. This driver setup file `teaboard.inf` is located in the common software folder (`common\soft`) on the Teaboard CD-ROM. (Figure 1.2)



Figure 1.2 Select Driver Setup File

3. Install Driver

Wait while Windows installs the driver. If a warning of “Windows Logo testing” appears, ignore it and click on **Continue Anyway**.

Wait until the installation is complete. Then click **Finish**.

In case of Windows Vista and Windows 7

1. Open “Device Manager” (“Control Panel” – “System and Security” – “System” – “Device Manager”).
2. On the Teaboard DIP switch DSW2, set both positions 1 and 2 to ON. Connect Teaboard and the Windows PC with a USB cable.
3. Double click on the “Unknown Device” (in “Other Devices” in Device Manager) to show its property.
If there are two or more unknown devices, select the unknown device with hardware ID “USB\VID_17EE&PID_0001”.
4. Click on “Update Driver” on the property of the unknown device.
5. To the query of searching folders, specify the folder which the driver setup file `teaboard_vista.inf` is stored. This driver setup file is stored in the common software folder (`common\soft`) on the Teaboard CD-ROM.
6. To the warning of certification, select the install button.

7. Click on the **Close** button at the end of the installation.

(2) Confirm the COM port number

Open the “Device Manager”, and select the “PMC Teaboard” in “Modem”.
(Figure 1.3)

† The Device Manager can be found by opening the Windows XP Control Panel (for example, from the Start menu), selecting “Performance and Maintenance,” opening the System applet, and looking under the Hardware tab in System Properties.

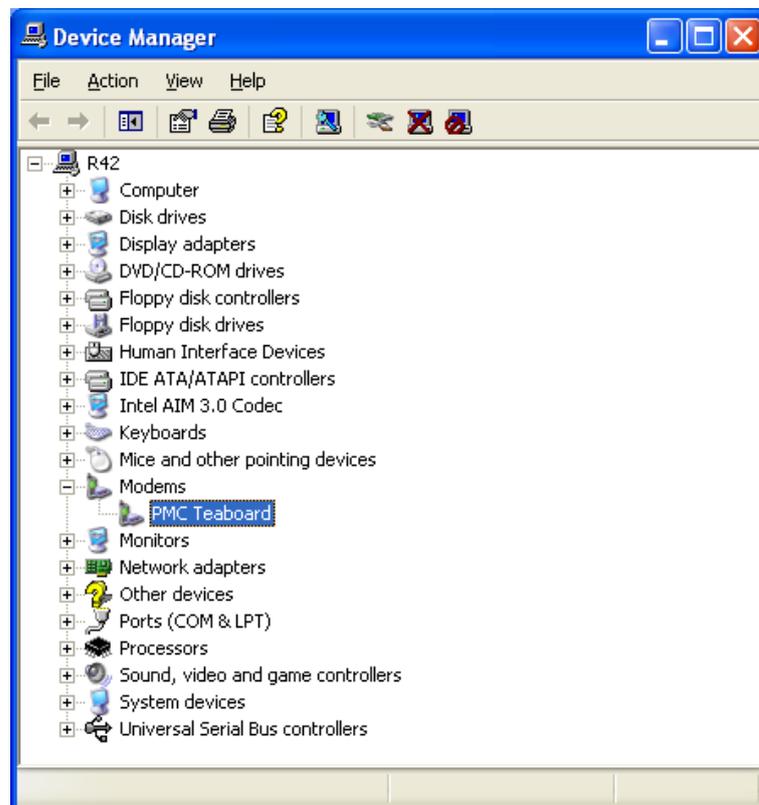


Figure 1.3 Opening Device Manager

Right-click to open its properties, open the “Modem” tab, and determine the COM port number. (Figure 1.4)

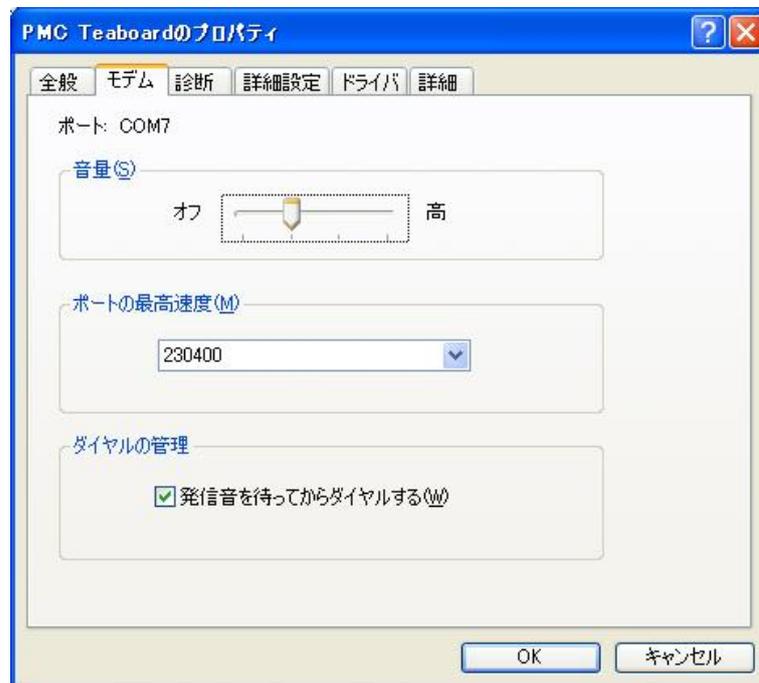


Figure 1.4 Confirm the COM port number

(3) Start Terminal Software

Now that the COM port is known, we can connect to that port using a terminal software. Any terminal program may be used. Here we use Hyper Terminal, but other software (such as Tera Term Pro) can be used.

† In cases of Windows Vista and Windows 7, 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.

Start Hyper Terminal, select the COM port number which you confirmed in (4), and set the communication parameters as follows:

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

† 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.).

(4) Confirm the Communication

Try pressing \leftrightarrow (Enter) Key a number of times on the terminal software. If a message from Teaboard appears on the terminal software (the T-Monitor prompt `TM>` etc.), connection was successful. (Figure 1.5)



Figure 1.5 T-Monitor prompt on a terminal software

(5) DIP-switch Setting

After closing the terminal software, disconnect the USB cable from Teaboard.

Make sure power is not being supplied to Teaboard, then return DIP switch DSW2-1 to OFF (it means booting T-Kernel), while DSW2-2 should remain ON (it means USB connection).

(6) Connecting USB Next Time

Note that the USB connection becomes disconnected when you boot/reset Teaboard. Before boot/reset Teaboard, you have to terminate your terminal software. And then boot/reset Teaboard, and restart your terminal software.

In case of the latest T-Monitor/T-Kernel and latest `gterm`, however, you do not have to terminate `gterm`, because the connection is kept even on resetting/rebooting Teaboard. To make this function enable, please update both your host-side development environment (including `te_vcom` and `gterm`) and the ROM image of Teaboard (including T-Monitor/T-Kernel). And please remake your boot disk.

Please note that not all terminal software support this function, while the latest `te_vcom` and `gterm` in the latest host-side development environment support it.

1.2 Installing Cygwin

Cygwin is required for the Teaboard Development Environment on PC.

For more details please also refer to the “*Cygwin Install Manual*” in the Teaboard CD-ROM.

(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 the product CD-ROM.

Cygwin System Package	<code>common\soft\cygwin.1.5.25-15.zip</code>
-----------------------	---

Expand this zip archive by Windows Explorer.

† Version numbers may be different.

† You can download the latest version of Cygwin from website (<http://www.cygwin.com/>) instead of the Cygwin System Package in the Teaboard CD-ROM. It is especially recommended in cases of Windows Vista and Windows 7. For more details please refer to the “*Cygwin Install Manual*” in the Teaboard CD-ROM.

(2) Start the Installer

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

Click on .

(3) Select Install Type

Select “Install from Local Directory” and click on .

(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 1.6)

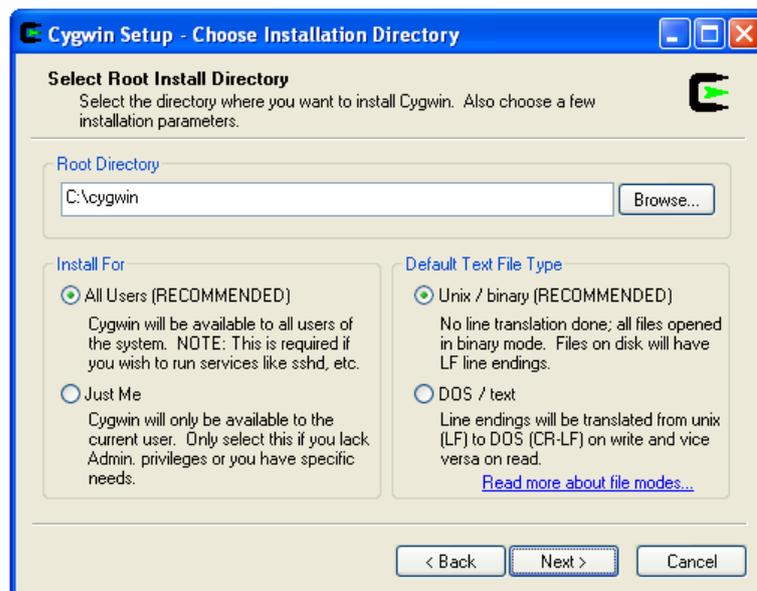


Figure 1.6 Select Install Directory

Select All Users at “Install For” and select “Unix / binary” as the “Default Text File Type”. Both are the default choices.

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 a file check is performed, a screen for selecting packages to install is displayed. The initial selected state is “All Default”. Click on “Default” to change to “Install”. (Figure 1.7)

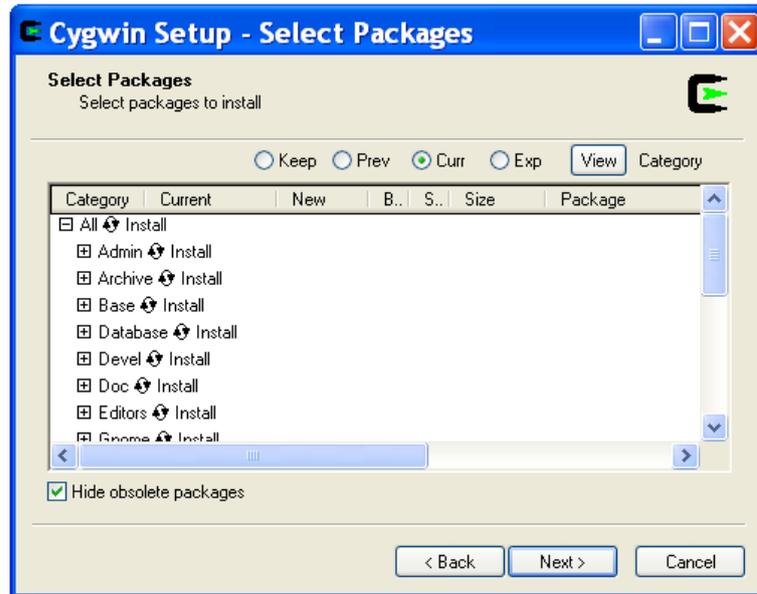


Figure 1.7 Select Install Package

(7) Install

Then click on 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.

The Cygwin installation is complete. Click on to 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
```

. Note that this can be done as the same login user at Cygwin installation.

(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↵
```

1.3 Install Eclipse and Teaboard Development Environment

Install Eclipse and add plug-ins for Teaboard Development.

For more details please also refer to the “*Eclipse Install Manual*” and the “*Eclipse-based T-Kernel Development Environment Install Manual*” in the Teaboard 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 Teaboard CD-ROM.

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

† 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 1.8). On the “Select a workspace” dialog, click on **Cancel** to quit Eclipse (Figure 1.9).

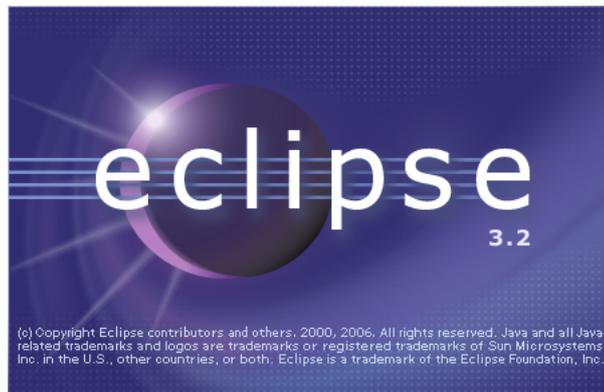


Figure 1.8 Starting Eclipse

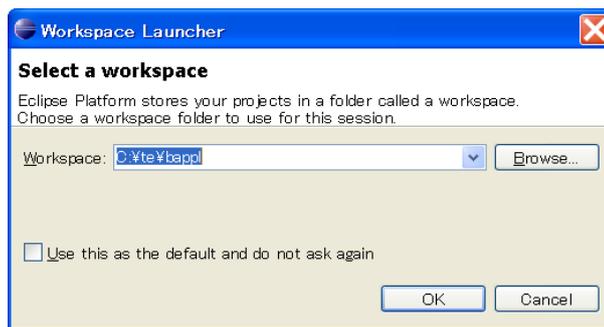


Figure 1.9 Select Workspace

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

(5) Expanding plug-ins for Teaboard Development

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 Teaboard CD-ROM.

Common part for Eclipse platform	<code>en\soft\com.t_engine4u.te.1.0.6.zip</code>
Machine-dependent part	<code>en\soft\com.t_engine4u.tl.tbmx1.1.0.2.zip</code>

† Version numbers may be different.

And then expand the archive files above to the Eclipse directory (such as `C:\eclipse`).

1.4 Initial Configuration

Configure the Development Environment.

- (1) Boot the target side (Teaboard)
 - Connect Teaboard with PC by USB.
- (2) Start Eclipse
 - Double-click on the shortcut of Eclipse to launch Eclipse.
- (3) 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.

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

- (4) Go to the Workbench
 - Then appears the “Welcome” view. Click on the icon of “Go to the workbench”. (Figure 1.10)

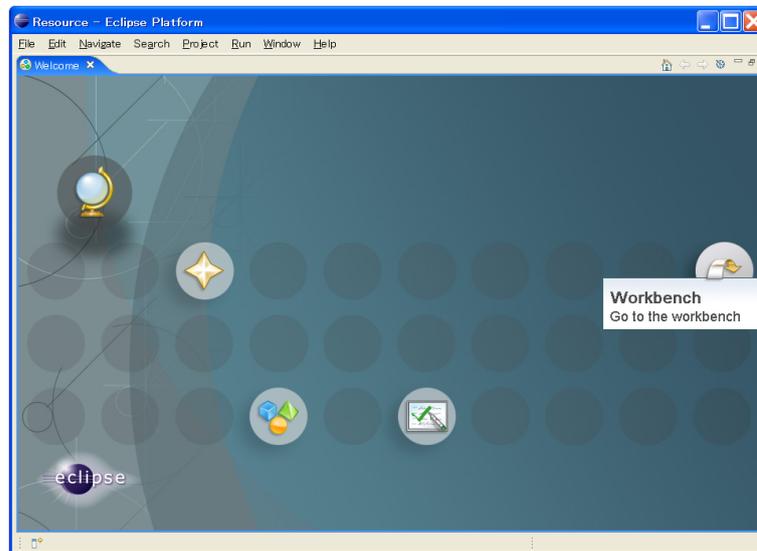


Figure 1.10 “Welcome” view of Eclipse

- (5) Open the “T-Engine Development” Perspective
 - Select Toolbar → “Window” → “Open Perspective” → “T-Engine Development”.
- (6) Setting of T-Engine Development
 - Select Toolbar → “Window” → “Preferences...” to open the Preferences Dialog. Select the page of “T-Engine Development”. (Figure 1.11)

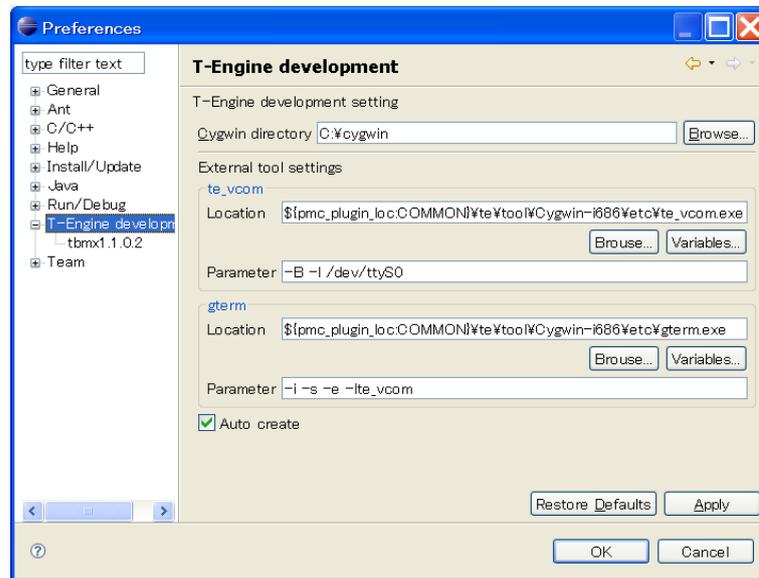


Figure 1.11 T-Engine Development Settings

As the parameter of `te_vcom`, specify the value of “COM port number of Teaboard minus one”. For example, specify “`-B -1 /dev/ttyS6`” for COM7.

† -1 is minus ell.

(7) Setting of Workspace

Select the page of “General” → “Workspace”. (Figure 1.12)

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

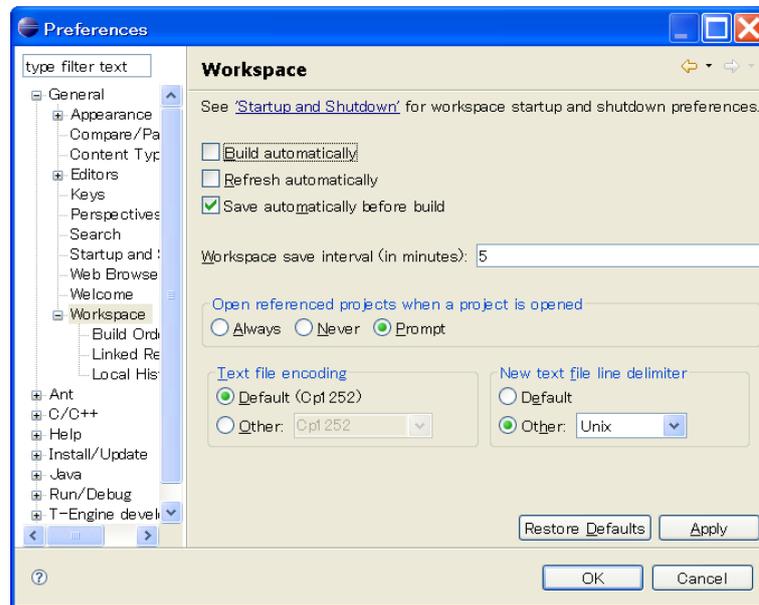


Figure 1.12 Setting of Workspace

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

(8) Opening console

Click on the “External Tools” button of Toolbar, and select “External Tools...” in the pull-down menu. (Figure 1.13)

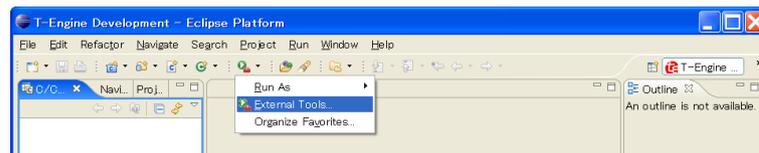


Figure 1.13 External Tools

And double-click on the “Program”. Then appear te_vcom and gterm under “Program”.

Select te_vcom and click on the Run button. (Figure 1.14)

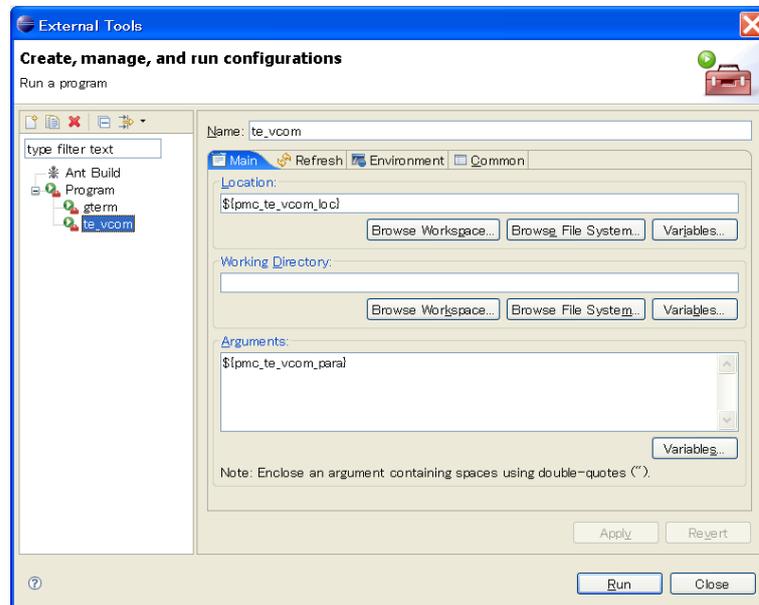


Figure 1.14 Running te_vcom

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 (Teaboard). This is a prompt of CLI (Command Line Interpreter), which runs on Teaboard. (Figure 1.15)

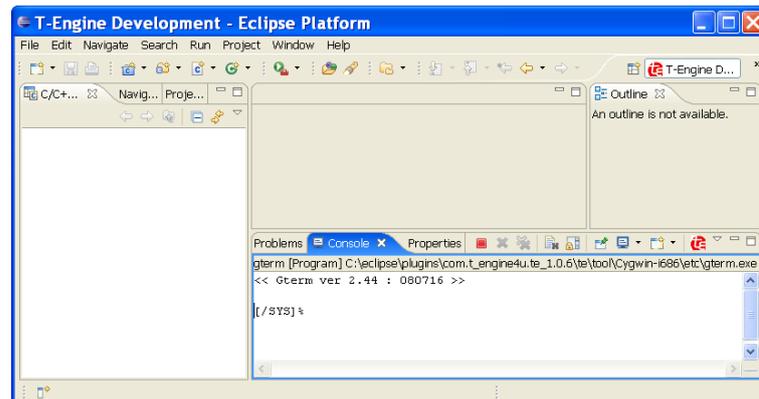


Figure 1.15 CLI prompt on gterm

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

1.5 How to Start Next Time

Start Teaboard and Eclipse next time as follows.

(1) Boot the target side (Teaboard)

At first connect Teaboard with PC by USB. (Do not start `te_vcom` and `gterm`.)

(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, `C:\te\kappl` for T-Kernel-based programs, and `C:\te\driver` for device drivers.

(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, you also have to run `te_vcom` and `gterm`.

(4) Confirm CLI Prompt

On the console window of `gterm`, hit `↵` (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`) from the target side (Teaboard).

† **Note on reset Teaboard**

In case of old version software, the USB connection becomes disconnected when you reset Teaboard (e.g. by pushing reset switch). Before reset Teaboard, you have to terminate `gterm` and `te_vcom` by the Terminate button (red square button). And then reset Teaboard, and restart `te_vcom` and `gterm` again.

The latest version software fixed this problem.

1.6 Boot from SD Card

Teaboard incorporates a Flash ROM, which serves as a ROM disk containing T-Kernel, configuration settings, tools, etc. Being a ROM disk means it cannot normally be written to.

† It is possible, however, to overwrite the contents using the T-Monitor command `writeRda`; see the *“Implementation Specification”* at 2, *“T-Monitor Implementation Specification”* for details.

Copying the ROM disk contents to an SD Card and booting from the SD Card is therefore a useful approach, one that allows configuration settings to be modified. Programs that are created for practice are likewise transferred to an SD Card and run from there.

The specifics are given in the *“Teaboard Manual”* in sections 3.1 and 3.2. The procedure is outlined below. It uses `gterm` and CLI.

(1) Make a partition on SD Card

Insert a blank SD Card, create partitions on it using the `hdpart` utility, and set a boot sector. The physical device name of SD Card is `pcb`.

```
[/SYS]% hdpart pcb↵ — make partition on SD Card (pcb)
pcb [C:485 H:32 S:32 B:497664 (243 MB)]
```

```

No System   Boot StartCHS   EndCHS   SecNo  SecCnt  Size
1 06 DOS     00  0:  1:  1  485: 31:32  32  497632 242 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] (<243MB) ? a↵      — partition size (all)
No System   Boot StartCHS   EndCHS   SecNo  SecCnt  Size
1 13 BTRON   00  0:  1:  1  485: 31:32  32  497632 242 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  485: 31:32  32  497632 242 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]%

```

- (2) Create a system disk on the 1st partition

Use the “format” utility to format the 1st partition on the SD Card. And copy all the ROM disk files to the SD Card.

The logical device name of the 1st partition is `pcb0`.

```

[/SYS]% format -b pcb0 system↵          — format 1st partition
Format pcb0 [STD] system
Logical Formatting...
Writing BootCode...
Disk Format Success.
[/SYS]% att pcb0 A↵                    — attach 1st partition
pcb0 -> pcb0
[/SYS]% rcp -b -r /SYS /A/=↵          — copy all files
[/SYS]% det pcb0↵                      — detach 1st partition
[/SYS]%

```

- (3) Confirm booting from the SD card

Push the reset button on Teaboard to reboot the system.

† In case of old version software, the USB connection becomes disconnected by reset. Before reset Teaboard, you have to terminate `germ` and `te.vcom` by the Terminate button (red square

button). And then reset Teaboard, and restart `te_vcom` and `gterm` again.

The latest version software fixed this problem.

Use the CLI command “`df`” to confirm that the system was booted from the 1st partition on the SD Card (logical device name is `pcb0`).

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

```

[/SYS]% df↵
PATH  DEV      TOTAL      FREE USED   UNIT MAXFILE NAME
/SYS  pcb0     248826     246564  0%  4096  21504 system
[/SYS]%

```

The above procedure results in creation of a boot disk on an SD Card. Thereafter, when Teaboard is started with the boot disk inserted, it will boot from this disk. Use the disk for further practice with Teaboard.

2 Development

2.1 Console Tools

Teaboard 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]% the CLI prompt changes to the mount name of whatever file system is attached). See the *“Development Tool Manual”* at 3, “Command Line Interpreter (CLI)” for details.

Utilities on CLI

Various utilities, such as “**hdpart**” (partitioning SD cards) and “**format**” (format SD cards) 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.” Teaboard 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 “oh” command.

```
TM> ? oh↵
OutputHalf(OH) port,data : Output Half to I/O port
```

(8) I/O write

Use the T-Monitor command “oh” to write the value 0 at address 0x16100000. Since this address connects to the 7-segment LED on the right, the entire LED will be lit. See the *Hardware Specification* at 2.8, “7-Segment LED” for an explanation of 7-segment LED control.

```
TM> oh 16100000, 0↵
Port 16100000:H <-- 0000
```

Display the pattern “3” on the right 7-segment LED. This can be done by extinguishing segment f (0x02), segment e (0x08), and the decimal point (0x20) while leaving the remaining segments on. Write the value 0x02 + 0x08 + 0x20 at address 0x16100000.

```
TM> oh 16100000, 02 + 08 + 20↵
Port 16100000:H <-- 002A
```

(9) Return from T-Monitor

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

```
TM> g↔
[/SYS]%
```

2.2 Process-Based and T-Kernel-Based Programs

Programs that run on Teaboard 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.

As memory space, an independent local space is assigned to each 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 Teaboard CD-ROM.

- “*T-Kernel Extension Manual*”
Process, tasks, files, events and so on.
- “*TCP/IP Manager Manual*”
Network (TCP/IP) functions.
- “*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 Teaboard 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.

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

Specify the following items. (Figure 2.1)

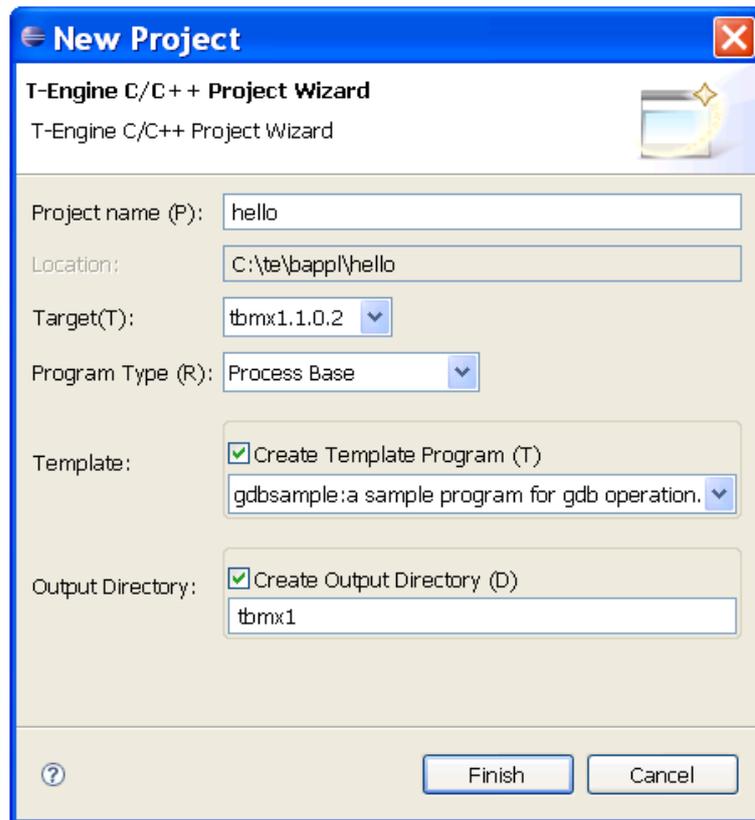


Figure 2.1 New Project Dialog

- Project name :
Specify your project name, for example, `hello`.
- Target :
Specify your target machine.
- 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/tbmx1/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/tbmx1/hello` is generated (with a human running icon).

[Trouble Shooting] If appears a message “Error launching builder”, confirm that you have selected the make target `hello/tbmx1/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/tbmx1/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 2.2)

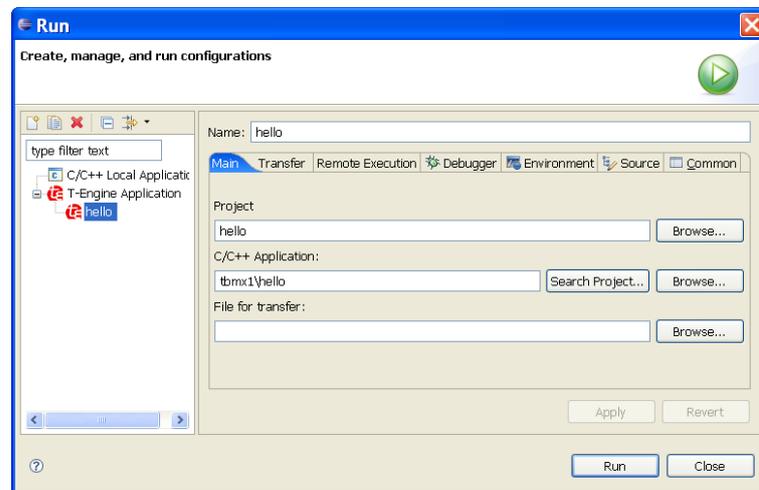


Figure 2.2 Run Dialog

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 2.3)

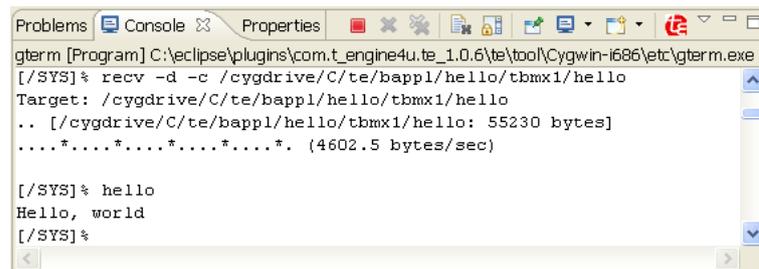


Figure 2.3 “Hello, world” appears on console

[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 Teaboard is booted from writable SD Card.

2.4 “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 **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 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 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.
- 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/tbmx1/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/tbmx1/hello2` is generated (without a human running 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/tbmx1/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] - 0x8026e000 - 6  rsdrv
[ 2] - 0x80282000 - 7  netdrv
... (omitted) ...
[ 5] - 0x80289000 - 7  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 (5 in the above example). Then appears “Hello, world” on the gterm console.

```
[/SYS]% unlspg 5↵
See you again
[/SYS]%
```

2.5 Running the Learning Programs

The Tutorial page on the Teaboard CD-ROM contains a collection of learning programs, archived as `exercise.zip`. Expand it on `C:\te` by Windows Explorer.

Details of each program are described in:

TRONWARE VOL. 95 (Personal Media Corporation; in Japanese)

See the following site for more about *TRONWARE*.

<http://www.personal-media.co.jp/book/>

2.5.1 Roulette

This is a T-Kernel-based program, which flashes the 7-segment LEDs to create a roulette-like effect.

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

Specify the following items.

- Project name :
Specify the project name, `roulette`.
- Target :
Specify your target machine.
- 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 has been extended.

- Output Directory:
Check it.

And click on the button to create a new project.

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

(3) Make (compile and link)

Select the target Makefile, `roulette/tbm1/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 `roulette/tbm1/roulette` is generated (without a human running icon, but it is correct).

(4) Transfer and Execute the Executable File

Select the executable file `roulette/tbm1/roulette`, 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.

The program causes the outer 8 segments of the 7-segment LEDs to light in sequence. Pressing the right push switch SW3 alternates between flashing and stopped states. Pressing the left push switch SW2 ends the display.

```
roulette: Can't Loaded System Program -1
[/SYS]%
```

This program deliberately returns the error value `-1` to keep from becoming resident, so this message is correct.

Note that the program disables interrupts for the left and right push switches.

2.5.2 Timer

This is a process-based program that counts down the specified number of seconds, and sounds a buzzer when zero is reached. As an example of device driver use, it relies on the LED control driver for 7-segment LED display, and on the buzzer control driver for sounding the buzzer.

(1) Select a Workspace for Device Drivers

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

If you have already launched Eclipse and works on other workspace, switch workspace to “C:\te\driver”, 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 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 Toolbar → “New” → “T-Engine C/C++ Project”).

Specify the following items.

- Project name :
Specify the project name, `led`.
- Target :
Specify your target machine.
- Program Type:
Specify “Driver”, or “Decide by workspace”, if your workspace is `C:\te\driver`.
- Template:
Do not check it, because the source program has been extended.
- Output Directory:
Check it.

And click on the button to create a new project.

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

(3) Make (compile and link)

Select the target Makefile, `led/tbmx1/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 `led/tbmx1/led` is generated (without a human running icon, but it is correct).

(4) Transfer and Execute the Executable File

Select the executable file `led/tbmx1/led`, 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.

† Now the LED driver is loaded, but it does not work until it is called by other application programs.

(5) Show the System Common Space

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

```
[/SYS]% ref spg↵
:
[ 5] - 0x8027b000 - 3 led
[/SYS]%
```

† System program ID and load address may be different.

(6) Create a Project, Make, and Execute the Buzzer Driver

Create a Project, Make, and Execute the Buzzer Driver “`buzzer`” in the same manner as LED driver.

Confirm that both `led` and `buzzer` are loaded on the system common space.

(7) Switch to a Workspace for Process-based Programs

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 window of `gterm`, hit ↵ (Enter) key several times, and confirm a CLI prompt (such as `[/SYS]%`).

(8) 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 the project name, `timer`.
- 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 has been extended.
- Output Directory:
Check it.

And click on the button to create a new project.

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

(9) Make (compile and link)

Select the target Makefile, `timer/tbmx1/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 `timer/tbmx1/timer` is generated (with a human running icon).

(10) Transfer and Execute the Executable File

Select the executable file `timer/tbmx1/timer`, 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.

The timer counts down for 60 seconds, and sounds a buzzer when zero is reached.

† Note on the buzzer:

- In case of Teaboard2/ARM920-MX1:
The buzzer sounds by connecting the hardware timer Timer2.
- In case of Teaboard/ARM920-MX1:
The buzzer sounds by GPIO D pin 31. The program should maintain the buzzer ON state for a least 100 ms.

See the “*Hardware Specification*” at 2.18, “Buzzer” for details.

2.5.3 Tiny Web Server

A tiny process-based Web server is provided as an example of a network application using TCP/IP.

(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 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 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 Teaboard.

```

[/SYS]% netconf c ↵
hostname = ? teaboard↵ — 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]%

```

(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 Teaboard 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 (Teaboard), 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 (Teaboard).

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 :
Specify 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 has been extended.
- Output Directory:
Check it.

And click on the button to create a new project.

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

(6) Make (compile and link)

Select the target Makefile, `webserv/tbmx1/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/tbmx1/webserv` is generated (with a human running icon).

(7) Transfer and Execute the Executable File

Select the executable file `webserv/tbmx1/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 button at the bottom of the Run Dialog to transfer and execute the program.

(8) View from Web Browser

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

(9) 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 2.4)



Figure 2.4 Send \^C on Eclipse

Index

	B		T-Monitor	21
buzzer		34	TCP/IP	34
	C		Tera Term Pro	8
CLI		21	terminal software	8
console		21	timer	31
Cygwin		9		U
	D		unlspg command	23
Device Manager		7		W
df command		20	web server	34
	E		Windows 7	6
Eclipse		12	Windows Vista	6
	F		Windows XP SP3	5
format command		19	workspace	14
	H			
hdpart command		18		
Hyper Terminal		8		
	I			
IMS		21		
	J			
Java Runtime		12		
	L			
LED		30		
lodspg command		23		
ls command		22		
	N			
netconf command		35		
	P			
Perl		12		
ping command		35		
process-based		23		
Project		32		
project		24		
	R			
recv command		35		
reset		18		
roulette		30		
	T			
T-Kernel-based		23		

Getting Started with Teaboard (on Eclipse)
Version 1.A0.05

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

Copyright © 2005–2010 by Personal Media Corporation
