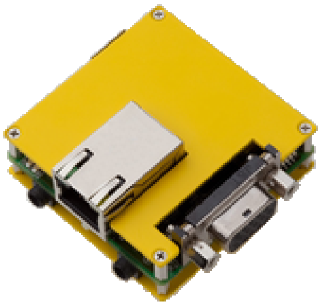
Getting Started with Teamacaron



Version 1.A0.00

Personal Media Corporation

Contents

History of Changes

Version 1.A0.00

- First edition for Teamacaron Development Kit.

Introduction

This document provides a tutorial on starting out with Teamacaron Development Kit, for those using this product for the first time.

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 Teamacaron Development Kit, we also use this style for software development on Teamacaron.

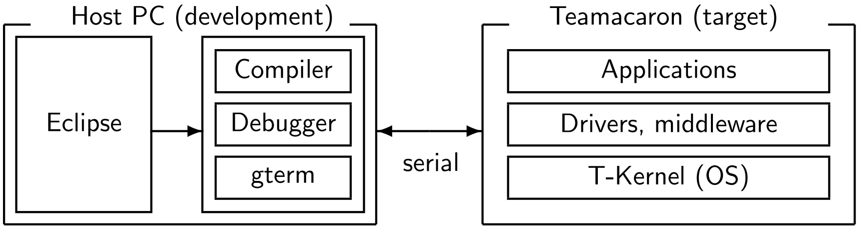


Figure 1.1 Cross-development

In this section we'll install the target-side environment (OS, drivers and middleware) on a microSD for Teamacaron, while the section 2 describes the installation For details please also refer to *"Teamacaron Development Kit Manual"* at "2. Installation".

1.1 System Installation

(1) Prepare microSD

Prepare a microSD card, and format it by Windows PC in FAT format.

† SHDC card is not supported. Instead please prepare a microSD card. And please format in FAT file system (not FAT32 file system).

And copy the following 3 files from the "`en\soft`" folder of the Teamacaron Development Kit CD-ROM to the root of the microSD.

File name	Content
`sdboot.bin`	boot loader
`tmonitor.bin`	T-Monitor
`romdisk.bin`	ROM disk image for installation

(2) Serial Connection

Connect your host PC and Teamacaron (UART0 connector) by serial connection (RS-232C). Please use the serial conversion cable included in the Teamacaron Development Kit(Figure 1.2). Please use USB-serial adapter, if your host PC has no serial (COM) port.

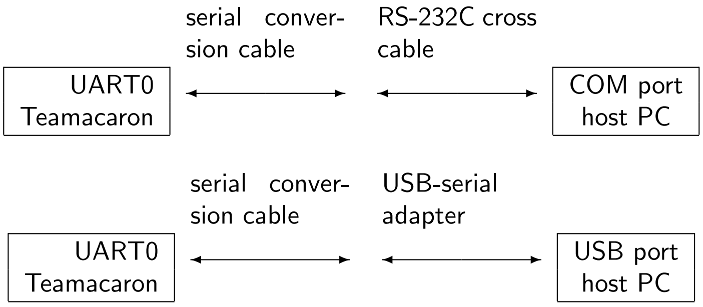


Figure 1.2 Serial cable connection

In both cases the conversion cable must be connected to the UART0 connector (CN5) of Teamacaron(Figure 1.3).

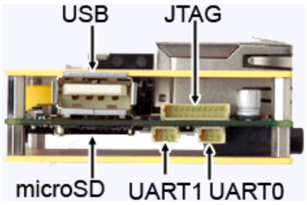


Figure 1.3 UART0 and microSD connectors

○ 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 :**

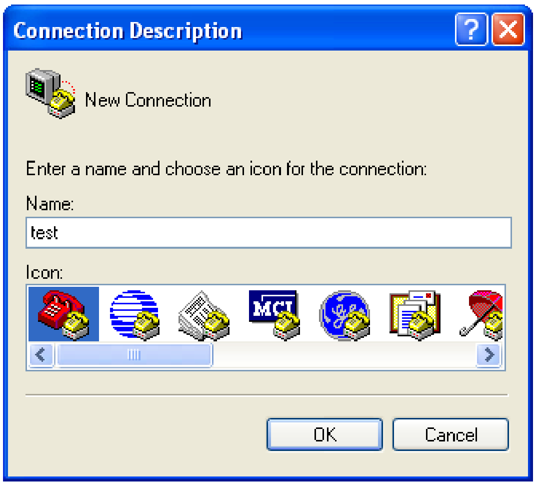
  Start Tera Term Pro.

- **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 OK. (Figure 1.4).



Enter a name and choose an icon for the connection:

Name: test

Figure 1.4 Start HyperTerminal

○ Communication Setting

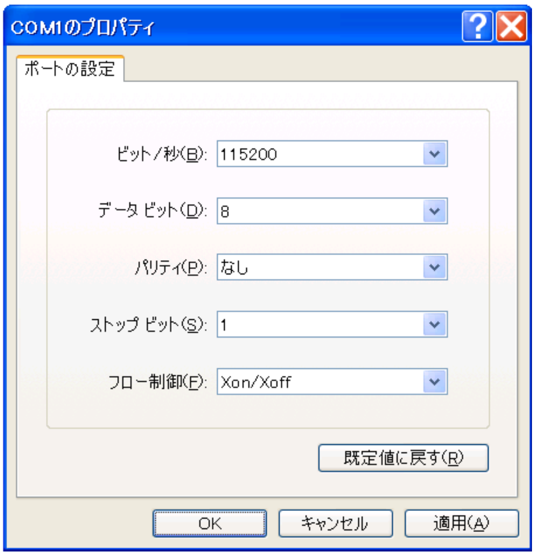
Select serial port of the host PC (COM1 etc.), and set the communication parameters as follows:

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 OK. Then select "Setup" → "Serial port", and set parameters as above.

- **In case of HyperTerminal :**
  Select the serial port (ex. COM1) and click OK. Then set the communication parameters (Figure 1.4).



ビット/秒(B): 115200

データ ビット(D): 8

パリティ(P): なし

ストップ ビット(S): 1

フロー制御(F): Xon/Xoff

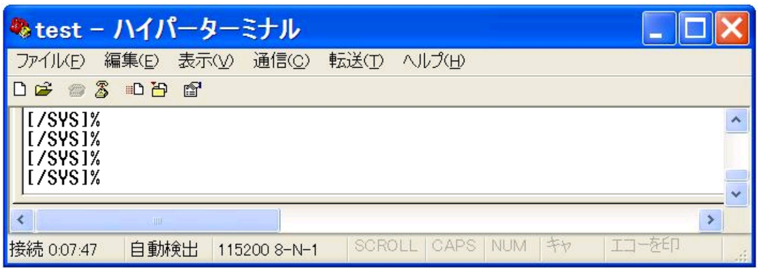
Figure 1.5 Communication Setting

(3) Boot Teamacaron

Insert the microSD to Teamacaron (Figure 1.3). And supply power to Teamacaron by AC adapter.

○ Confirming Prompt

Try pressing ↩ (Enter) Key a number of times on the terminal software. The CLI prompt (`[/SYS]%`) will appear on the terminal software (Figure 1.6).


test – ハイパーターミナル
ファイル(F) 編集(E) 表示(V) 通信(C) 転送(T) ヘルプ(H)

```
[/SYS]%
[/SYS]%
[/SYS]%
[/SYS]%
```

接続 0:07:47
自動検出
115200 8-N-1
SCROLL
CAPS
NUM
キャ
エコーを印

Figure 1.6 CLI prompt on a terminal software

(4) Making Two Partitions

Run `hdpart` command from the terminal software, where the device name is "pcb", which means the whole disk. Create two partitions, where the 1st partition must be more than 32 MB. And set 2nd partition as the boot partition.

```
[/SYS]% hdpart pcb↩                              — Run hdpart command
pcb    [C:955   H:64   S:63   B:3862528 (1886 MB)]
No System     Boot  StartCHS      EndCHS     SecNo   SecCnt     Size
1   06 DOS     00     0:   2:10   956:  63:63    135  3858489  1884 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] (<1886MB) ? 32MB↩            — Partition size
No System     Boot  StartCHS     EndCHS    SecNo  SecCnt     Size
1  13 BTRON   00    0:  1: 1   16: 63:63     63    68481   33 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 ? c↩          — Create partition
Create PartNo (1-4) ? 2↩                           — 2nd partition
Size [GB/MB/KB,All] (<1853MB) ? a↩                  — All the rest
No System     Boot  StartCHS     EndCHS    SecNo  SecCnt     Size
1  13 BTRON   00    0:  1: 1   16: 63:63     63    68481   33 MB
2  13 BTRON   00   17:  0: 1  957: 61:61  68544 3793984 1852 MB
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) ? 2↩                       — 2nd partition
No System     Boot  StartCHS     EndCHS    SecNo  SecCnt     Size
1  13 BTRON   00    0:  1: 1   16: 63:63     63    68481   33 MB
2  13 BTRON   80   17:  0: 1  957: 61:61  68544 3793984 1852 MB
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]%
```

(5) Format the 2nd partition

Format the 2nd partition by `format` command from the terminal software. Device name is "`pcb1`". And "`-b`" option (write boot record) is required. Also "`-x`" option is required if the partition size is 2GB or more.

```
[/SYS]% format -b pcb1 SYSTEM↩           — format 2nd partition
Format pcb1 [STD] SYSTEM
Logical Formatting...
Writing BootCode...
Disk Format Success.
[/SYS]%
```

(6) Copy the ROM disk files

Copy the all ROM disk files to the 2nd partition.

```
[/SYS]% att pcb1 A↩                      — Attach 2nd partition
hda0 -> A
[/SYS]% rcp -r /SYS /A/=↩                   — Copy basic system
```

```
[/SYS]% det pcb1↩                              — Detach 2nd partition
[/SYS]%
```

(7) Shut down Teamacaron

Use exit twice to shutdown the target.

```
[/SYS]% exit↩
<< EXIT cli >>
[IMS]% exit     1
[IMS]% exit↩
<< SYSTEM SHUTDOWN >>
```

(8) Format the 1st partition in FAT format

Eject microSD from Teamacaron, insert it to Windows PC, and format the 1st partition in FAT format.

† Please format in FAT file system (not FAT32 file system). Please also confirm that the partition capacity is nearly equal to the size of the 1st partition, which is created in (4) before.

(9) Copy files

Copy the following 5 files from the "en\soft" folder of the Teamacaron Development Kit CD-ROM to the root of the 1st (FAT) partition of the microSD.

File name	Content
sdboot.bin	boot loader
tmonitor.bin	T-Monitor
install.bz	Basic system install package
install_tshell.bz	T-Shell install package
english.bz	T-Shell english patch

† Do not copy romdisk.bin in this time.

(10) Boot Teamacaron

Insert the microSD card again to Teamacaron. And supply power to Teamacaron by AC adapter.

○ **Confirming Prompt**

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

(11) Installing the Basic System

```
[/SYS]% att -m pcb0 M↩                          — attach 1st partition
pcb0 -> M
[/SYS]% expf -U /M/install.bz↩              — expand install package
[/SYS]% install /SYS↩                                        — install
Install completed, please reboot the system.
[/SYS]% rm -r install↩                              — remove installer
```

(12) Installing T-Shell

```
[/SYS]% expf -U /M/install_tshell.bz↩  — expand install package
[/SYS]% install_tshell /SYS↩                           — install
Install completed, please reboot the system.
[/SYS]% rm -r install_tshell↩                — remove installer
[/SYS]%
```

○ Installing T-Shell English Patch

This English patch translates T-Shell GUI applications to English.

```
[/SYS]% expf -U /M/english.bz↩                  — expand patch
[/SYS]% english /SYS↩                                — install
..  Installing English Patch on /SYS
..  Install completed
[/SYS]% rm -r english↩                          — remove patch
[/SYS]%
```

○ Detach 1st partition

```
[/SYS]% det -u pcb0↩                   — detach 1st partition
[/SYS]%
```

(13) Reboot Teamacaron

Use “exit” and “exit -1” commands to reboot Teamacaron.

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

If you connect a display connected to the CRT connector of Teamacaron it shows an initial window of PMC T-Shell (Figure 1.7). And if you connect an USB mouse, your can move pointer or show right-click menu.

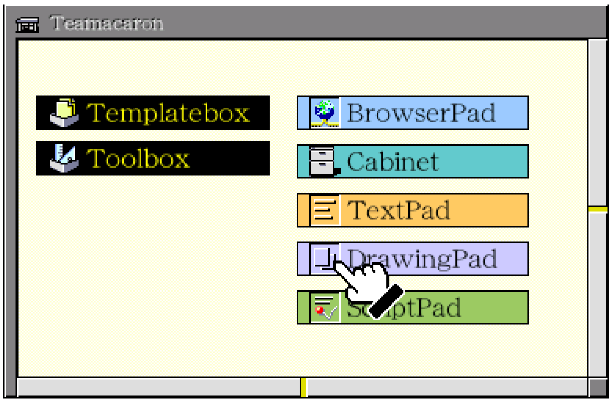


Figure 1.7 Initial Window of PMC T-Shell

On the other hand, the terminal software of the host PC shows booting messages of Teamacaron, and the CLI prompt (`[/SYS]%`).

○ **Screen Size**

Screen size can be changed as follows. The changed setting becomes valid after reboot.

```
[/SYS]% devconf VIDEOMODE 8↩
[/SYS]% exit↩
<< EXIT cli >>
[IMS]% exit     1
[IMS]% exit -1↩
<< SYSTEM RESTART >>
    ⋮
[/SYS]%
```

VIDEOMODE	size
5	640 × 480
8	800 × 600

1.2 Font Installation

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

The following install files are for English fonts:

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

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

(1) Shut down Teamacaron

Use `exit` twice to shutdown the target.

```
[/SYS]% exit↩
<< EXIT cli >>
[IMS]% exit     1
[IMS]% exit↩
<< SYSTEM SHUTDOWN >>
```

(2) Copy font file

Eject microSD from Teamacaron, and insert it to Windows PC. And copy the following 2 files from the "`common\font`" folder of the Teamacaron Development Kit CD-ROM to the root of the 1st (FAT) partition of the microSD.

File name	Content
`instfont`	font installer
`SSHELVET`	SSHelvetica font

(3) Boot Teamacaron

Insert the microSD card again to Teamacaron. And supply power to Teamacaron by AC adapter.

(4) Installing font

```
[/SYS]% att -m pcb0 M↩                          — attach 1st partition
pcb0 -> M
[/SYS]% /M/instfont /M/SSHELVET↩                       — install font
    Font File Name     : regist
 0: **-****            : use
 1: ****-SSHelvetica** : use
[/SYS]% det -u pcb0↩                            — detach 1st partition
[/SYS]%
```

(5) Reboot Teamacaron

Use “exit” and “exit -1” commands to reboot Teamacaron.

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

2 Installing Development Environment

† The development environment for Teamacaron Development Kit works on Windows XP, Windows Vista and Windows 7.

2.1 Installing Cygwin

Cygwin is required for the Teamacaron Development Environment on PC. For more details please also refer to the *"Cygwin Install Manual"* in the Teamacaron Development Kit CD-ROM.

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

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

(1) Expand Cygwin Package

Copy (save) the following file to the root of your install drive (such as `C:\`) from the common software directory (`common\soft`) of Teamacaron Development Kit CD-ROM.

Cygwin System Package	`common\soft\cygwin\_d.1.7.5-1.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)

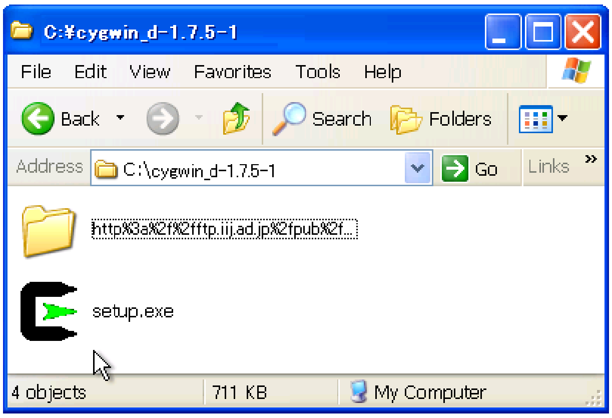


Figure 2.1 Start Cygwin Installer

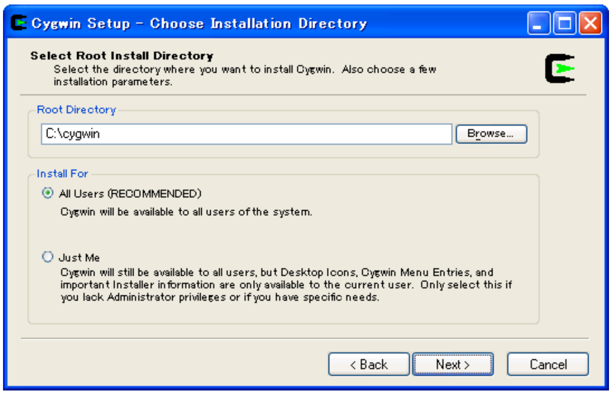
Click on Next >.

(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)



Cygwin Setup - Choose Installation Directory

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

Root Directory
C:\cygwin

Install For
● All Users (RECOMMENDED)
Cygwin will be available to all users of the system.

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

Figure 2.2 Select Install Directory

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

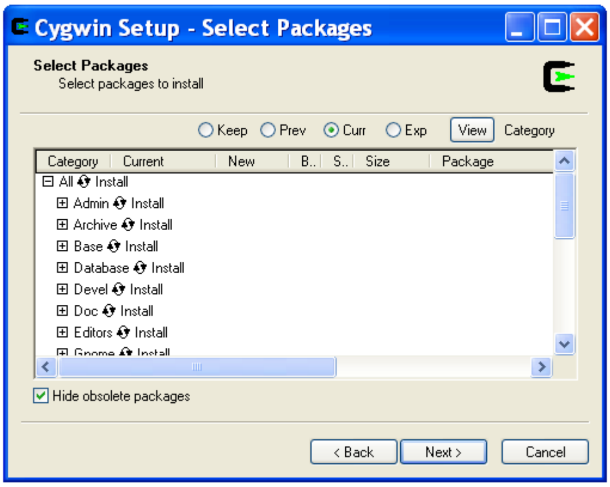
(5) Select Local Package Directory

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

Click on Next > to continue.

(6) Select Install Packages

After checking files, a “Setup Alert” dialog may appear. If there is no problem, click on OK. 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)



Cygwin Setup - Select Packages

Select Packages
Select packages to install

○ Keep ○ Prev ● Curr ○ Exp

Category	Current	New	B..	S..	Size	Package
⊟ All ♻ Install						
⊞ Admin ♻ Install						
⊞ Archive ♻ Install						
⊞ Base ♻ Install						
⊞ Database ♻ Install						
⊞ Devel ♻ Install						
⊞ Doc ♻ Install						
⊞ Editors ♻ Install						
⊞ Gnome ♻ Install						

☑ Hide obsolete packages

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 Teamacaron Development 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/`

(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 Teamacaron Development 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:\`).

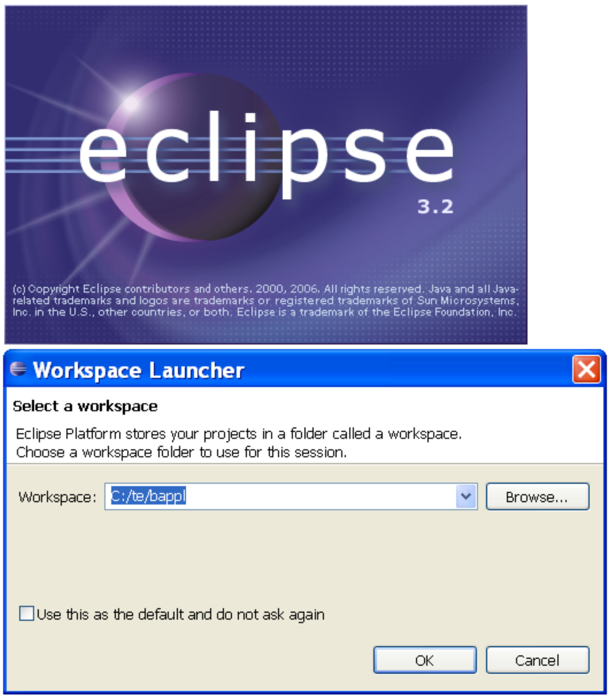
† If you expand the packages to `C:\`, the new folder `C:\eclipse` is created, and the packages are expanded under this folder.

(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.



(c) Copyright Eclipse contributors and others. 2000, 2006. All rights reserved. Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.

Workspace Launcher

Select a workspace

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

Workspace: C:/te/bappl

□ Use this as the default and do not ask again

Figure 2.4 Starting Eclipse

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

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 Teamacaron Development 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 Teamacaron Development Kit CD-ROM.

Common part for Eclipse platform	`com.t_engine4u.te.1.0.9.zip`
Machine-dependent part	`com.t_engine4u.tl.em1d512.1.0.0.zip`

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

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

† Version numbers may be different.

(2) Expand T-Shell Additional Package

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

T-Shell additional package	`com.t_engine4u.tsh.em1d512.1.0.0.zip`

And expand this archive to
`C:\eclipse\plugins\com.t_engine4u.tl.em1d512.1.0.0_1.0.0\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 OK.

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

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

(5) Go to the Workbench

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

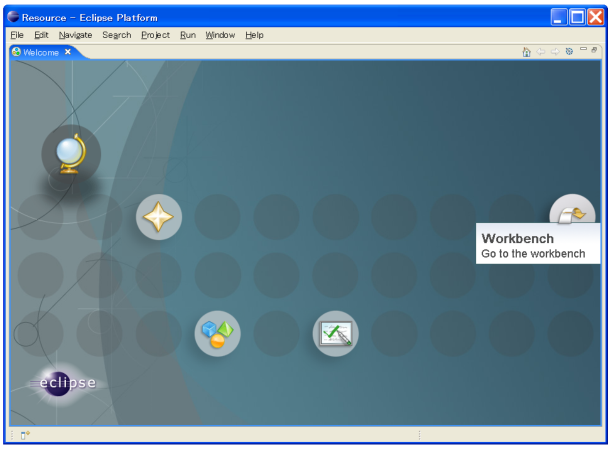

Resource - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Welcome
Workbench
Go to the workbench
eclipse

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)

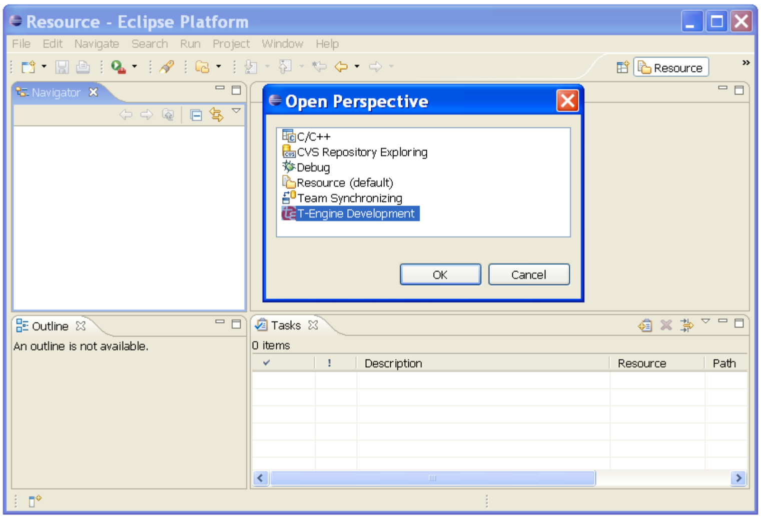
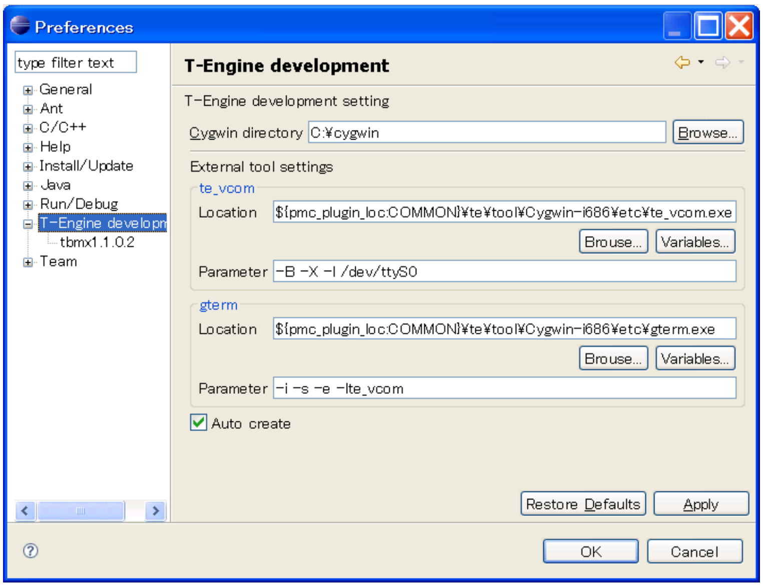


Figure 2.6 Select T-Engine development perspective

(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)



T-Engine development

T−Engine development setting

Cygwin directory C:¥cygwin

External tool settings

te_vcom

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

Parameter −B −X −l /dev/ttyS0

gterm

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

Parameter −i −s −e −lte_vcom

☑ Auto create

Figure 2.7 T-Engine Development Settings

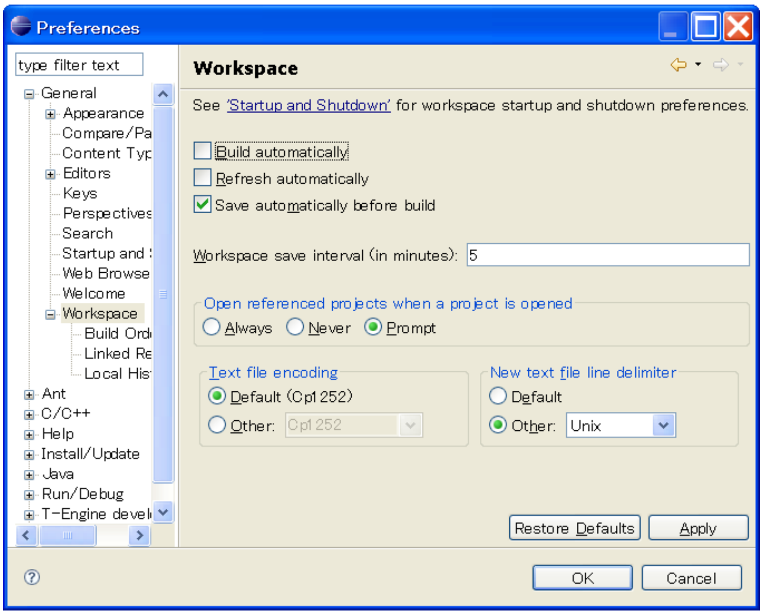
As the parameter of te_vcom, specify "`-B -X -l /dev/ttyS0`" in case of serial port COM1.

† "`-B`" means 115200 bps, and "`-X`" means suppressing hardware flow control (RTS/CTS). And `-l` is minus ell, stands for line device. Specify "`-B -X -l /dev/ttyS1`" for COM2, and so on.

(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.



Workspace

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

□ Build automatically

□ Refresh automatically

☑ Save automatically before build

Workspace save interval (in minutes): 5

Open referenced projects when a project is opened

○ Always ○ Never ● Prompt

Text file encoding

● Default (Cp1252)

○ Other: Cp1252

New text file line delimiter

○ Default

● 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)

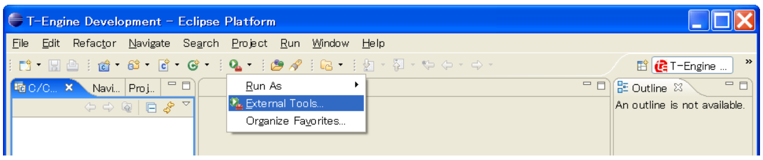
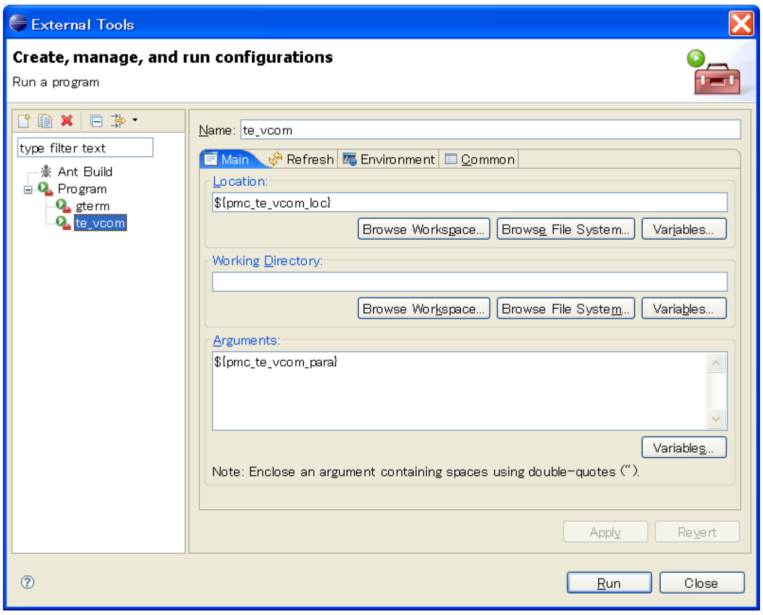

T-Engine Development - Eclipse Platform
File Edit Refactor Navigate Search Project Run Window Help
T-Engine ...
C/C... Navi... Proj...
Run As
External Tools...
Organize Favorites...
Outline
An outline is not available.

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)



Create, manage, and run configurations

Run a program

Name: te_vcom

Location:
${pmc_te_vcom_loc}

Working Directory:

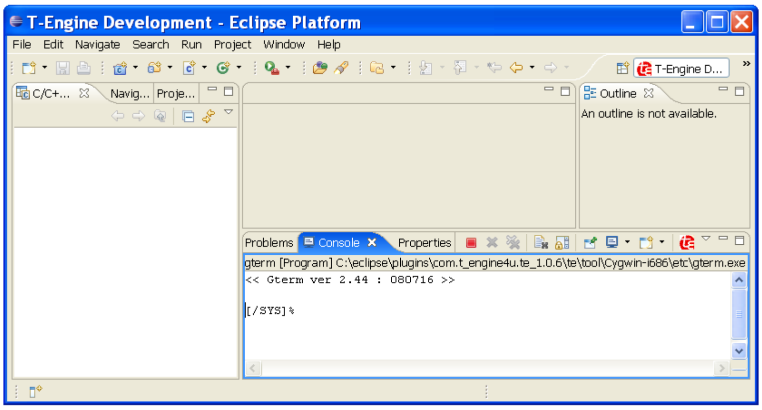
Arguments:
${pmc_te_vcom_para}

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

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 (Teamacaron). This is a prompt of CLI (Command Line Interpreter), which runs on Teamacaron. (Figure 2.11)



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

```
gterm [Program] C:\eclipse\plugins\com.t_engine4u.te_1.0.6\te\tool\Cygwin-i686\etc\gterm.exe
<< Gterm ver 2.44 : 080716 >>

[/SYS]%
```

Figure 2.11 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 Teamacaron and Eclipse as follows.

(1) Boot the target side (Teamacaron)

Supply power to Teamacaron by AC adapeter.

The display connected to the Teamacaron shows an initial window on PMC T-Shell.

(2) Start Eclipse

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

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

(3) Run te_vcom and gterm

Select the external tools, and run te_vcom and gterm.

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

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

† 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 Teamacaron (Figure 3.1).

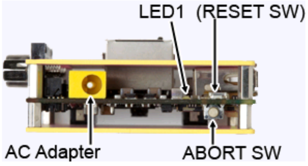


Figure 3.1 Reset switch

3.2 Console Tools

Teamacaron 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." Teamacaron 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 “ow” command.

```
TM> ?  ow↩
OutputWord(OW) port,data : Output Word to I/O port
```

(8) I/O write

Use the T-Monitor command “ow” to write 32 bit values to address 0xc005008c to control LED2 (red) and LED3 (orange) (Figure 3.2).

```
TM> ow c005008c,c0000000↩
Port C005008C:W <-- C0000000
TM> ow c005008c,c0004000↩
Port C005008C:W <-- C0004000
TM> ow c005008c,c0008000↩
Port C005008C:W <-- C0008000
TM> ow c005008c,c000c000↩
Port C005008C:W <-- C000C000
```

value	LED2	LED3
0xc0000000	ON	ON
0xc0004000	OFF	ON
0xc0008000	ON	OFF
0xc000c000	OFF	OFF

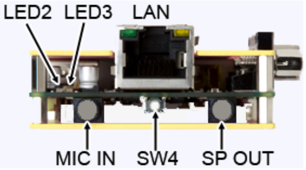


Figure 3.2 LED2 and LED3

(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 Teamacaron 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 Teamacaron Development 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 Teamacaron Development Kit CD-ROM.

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

- "*Library Manual*"
  Standard C library and other libraries.

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

- "*Device Driver Manual*"
  T-Kernel-based programs can call device drivers, as well as making device drivers.

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

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

(1) Select a Workspace

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

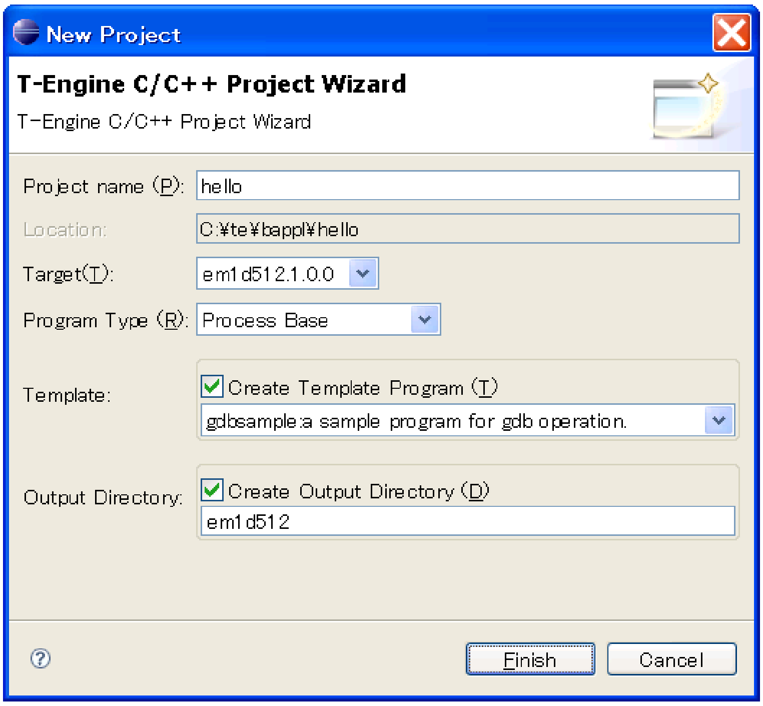
If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\bappl`", by Menubar → "File" → "Switch Workspace...".

† 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 3.3)



New Project

T-Engine C/C++ Project Wizard

T-Engine C/C++ Project Wizard

Project name (P): hello

Location: C:¥te¥bappl¥hello

Target(T): em1d512.1.0.0

Program Type (R): Process Base

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

Output Directory: ☑ Create Output Directory (D)
em1d512

Figure 3.3 New Project Dialog

- Project name :
  Specify your project name, for example, `hello`.
- Target :
  Specify your target machine (em1d512).
- 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 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/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/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.

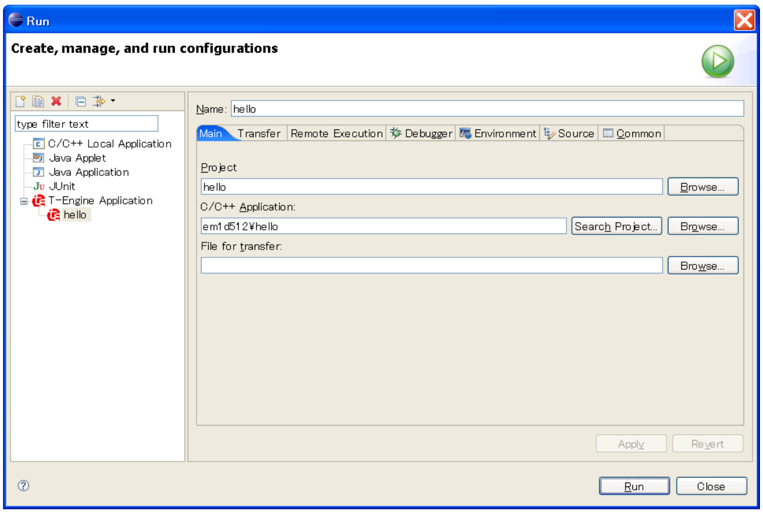
[**Trouble Shooting**] If appears a message "Error launching builder", confirm that you have selected the make target `hello/em1d512/Makefile`.

[**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/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 3.4)



Name: hello

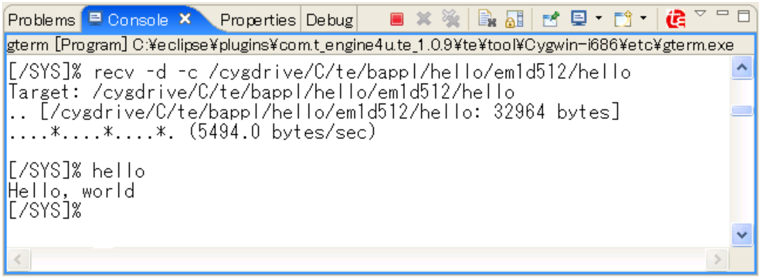
Project
hello

C/C++ Application:
em1d512¥hello

File for transfer:

Figure 3.4 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 3.5)



```
gterm [Program] C:¥eclipse¥plugins¥com.t_engine4u.te_1.0.9¥te¥tool¥Cygwin-i686¥etc¥gterm.exe
[/SYS]% recv -d -c /cygdrive/C/te/bappl/hello/em1d512/hello
Target: /cygdrive/C/te/bappl/hello/em1d512/hello
.. [/cygdrive/C/te/bappl/hello/em1d512/hello: 32964 bytes]
....*....*....*. (5494.0 bytes/sec)

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

Figure 3.5 “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 Teamacaron is booted from writable disk.

3.5 “Hello, world” as a T-Kernel-Based Program

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

(1) Select a Workspace

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

If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\kappl`", by 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).
- 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 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/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/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/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] - 0xd0186000 -  6  screen
··· omitted ···
[13] - 0xd0760000 -  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 on the Teamacaron Development Kit CD-ROM contains a collection of learning sample programs.

- **Juggling**

  This program shows juggling animation.

- **Tiny Web Server**

  This program transfers files from Teamacaron 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.

4.1 Juggling

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

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

(1) Select a Workspace

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

If you have already launched Eclipse and works on other workspace, switch workspace to “`C:\te\bappl`”, by 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).
- Program Type:
  Specify “Process Base”, or “Decide by workspace”, if your workspace is `C:\te\bappl`.
- Template:
  Do not check it, because the source program will be expanded later.

- Output Directory:
  Check it.

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

(3) Expand Source Program

The source program is archived in "`juggling.zip`" on the "`en\tutorial`" folder in the Teamacaron Development 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/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/juggling`" is generated (with a runner or bug icon).

(5) Transfer and Execute the Executable File

Select the executable file "`juggling/em1d512/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 Run button at the bottom of the Run Dialog to transfer and execute the program. Juggling animation is showed on the screen (Figure 4.1)

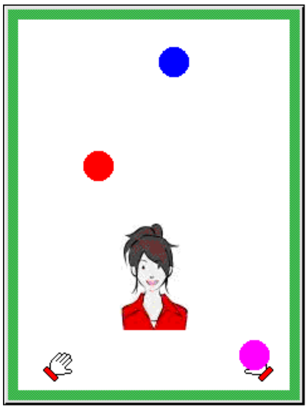


Figure 4.1 Run Juggling

(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.debug`".

Select the target Makefile, "`juggling/em1d512.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.debug/juggling`" is generated with debug information (showed by a insect icon).

(7) Start Debugging

Select the executable file "`juggling/em1d512.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.

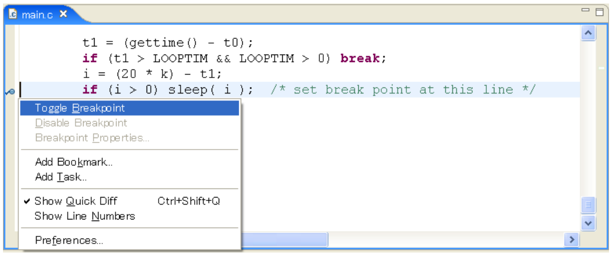
† There remains a previous setting created in (5) for execution. But create a new one for debugging.

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)

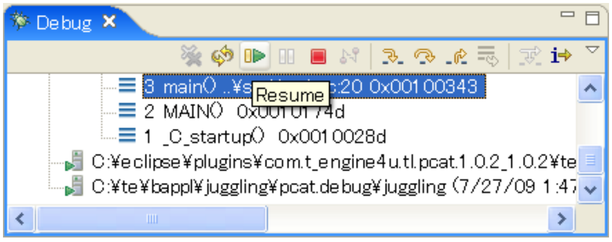


```
        t1 = (gettime() - t0);
        if (t1 > LOOPTIM && LOOPTIM > 0) break;
        i = (20 * k) - t1;
        if (i > 0) sleep( i );  /* set break point at this line */
```

Figure 4.2 Set a Breakpoint

(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)



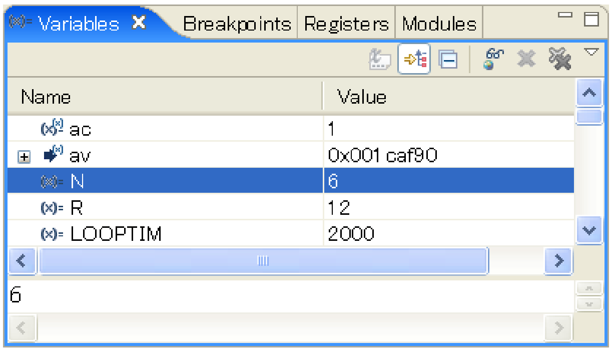
```
3 main() ..¥s        c:20 0x00100343
2 MAIN() 0x001  74d
1 _C_startup() 0x0010028d
C:¥eclipse¥plugins¥com.t_engine4u.tl.pcat.1.0.2_1.0.2¥te
C:¥te¥bappl¥juggling¥pcat.debug¥juggling (7/27/09 1:47
```

Figure 4.3 Continue Execution

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

(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)



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

6

Figure 4.4 Change a Variable

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

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

(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 Teamacaron are transfered to web browsers on other PCs.

(1) Select a Workspace

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

If you have already launched Eclipse and works on other workspace, switch workspace to "`C:\te\bappl`", by 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 Teamacaron.

```
[/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]%
```

† 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 ↩                  — pint another machine
192.168.0.1 is alive <192.168.0.1> : 0 ms
```

† In this example, the Teamacaron 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 (Teamacaron), 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 (Teamacaron).

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

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

(6) Expand Source Program

The source program is archived in “`webserv.zip`” on the “`en\tutorial`” folder in the Teamacaron Development 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/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/webserv`” is generated (with a runner or bug icon).

(8) Transfer and Execute the Executable File

Select the executable file "`webserv/em1d512/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 Teamacaron IP address in a Web browser, and confirm that the images loaded earlier are listed in the browser. If, for example, the Teamacaron IP address is 192.168.0.70, enter the URL as follows:

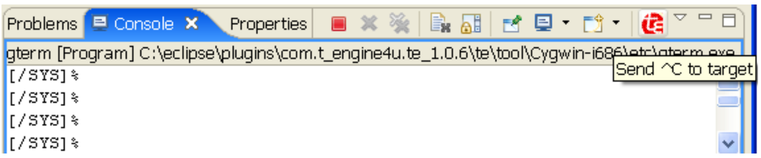
```
http://192.168.0.70/index.html
```

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

(10) Terminate Process

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

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



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

Figure 4.5 Send ˆC 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\kappl`" for T-Kernel-based programs, and click on OK.

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

† If you switched workspace, also terminate external tools te_vcom and gterm. Then you have to run te_vcom and gterm again. On the console 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).
- Program Type:
  Specify “T-Kernel Base”, or “Decide by workspace”, if your workspace is `C:\te\kappl`.
- Template:
  Do not check it, because the source program will be expanded later.
- Output Directory:
  Check it.

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

(3) Expand Source Program

The source program is archived in “`ledint.zip`” on the “`en\tutorial`” folder in the Teamacaron Development 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/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/ledint` is generated.

(5) Transfer and Execute the Executable File

Select the executable file `ledint/em1d512/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 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.

By this program LED2 (red) or LED3 (orange) start blinking. By pushing SW4 (push switch near the LAN connector) changes blinking pattern (Figure 4.6).

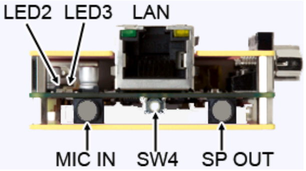


Figure 4.6 LED2, LED3 and SW4

Index

Getting Started with Teamacaron
Version 1.A0.00

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