

はじめてみよう Teaboard (Eclipse 環境編)

Version 1.A0.05

目次	2
----	---

目次

修正履歴	3
はじめに	4
1 セットアップ編	6
1.1 Teaboard とパソコンの USB 接続	6
1.2 Cygwin のインストール	12
1.3 Eclipse と Teaboard 開発環境のインストール	15
1.4 開発環境の初期設定	18
1.5 次回以降の起動	22
1.6 SD カードからの起動	23
2 開発編	26
2.1 コンソールの利用	26
2.2 プロセスベースと T-Kernel ベース	28
2.3 “Hello, world” プロセスベース編	30
2.4 「Hello, world」 T-Kernel ベース編	34
2.5 実習用プログラムの実行	37
2.5.1 ルーレット	37
2.5.2 タイマー	39
2.5.3 簡易ウェブサーバ	43
索引	47

修正履歴

Version 1.A0.05

- Windows Vista と Windows 7 について記載を追加しました。

Version 1.A0.04

- Windows Vista について記載しました。

Version 1.A0.03

- パソコンと Teaboard を USB 接続する場合、USB ハブを介さず直接接続することをおすすめします。
- パソコン と Teaboard を USB 接続する場合、最新バージョンのソフトウェアでは Teaboard のリセット時に USB 接続が切断されなくなったため、通信ソフト `te_vcom`, `gterm` を終了、再起動する必要はありません。

Version 1.A0.02

- `hdpart` コマンドによる区画作成時に、既に区画が存在する場合はいったん削除するように手順を修正しました。

Version 1.A0.01

- Cygwin と Eclipse を CD からインストールする方法に変更しました。
- 実習用プログラム集のアーカイブ形式を `zip` に変更しました。

Version 1.A0.00

- コンソールベースの開発環境の『はじめてみよう Teaboard』を元に、Eclipse 環境用に新規作成。

はじめに

本書では、はじめて Teaboard をご利用になる方を対象に、Teaboard を使い始めるためのチュートリアルをご提供します。開発環境としては Eclipse ベースの統合開発環境を使います。

† Teaboard の開発環境としては次の二種類が利用できますが、本書では Eclipse ベースの開発環境の場合で説明します。

- Eclipse ベースの統合開発環境：
主に Eclipse 上のメニューやダイアログの操作で、プログラムのメイク、転送、実行、デバッグなどの開発作業を行います。
- コンソールベースの開発環境：
Cygwin や Linux 上のコンソール上でコマンドを入力しながら開発作業を行います。

† Teaboard の開発環境は WindowsXP SP3、Windows Vista、Windows 7 に対応しています。

本書全体の流れは次の通りです。

• セットアップ編

1. Teaboard とパソコンの USB 接続

Teaboard とパソコンを USB で接続するために、Windows にデバイスドライバをインストールします。

2. Cygwin のインストール

Windows パソコン上で開発環境を使うには、まず Cygwin をインストールする必要があります。

3. Eclipse と Teaboard 開発環境のインストール

Windows 上に Eclipse をインストールして、Teaboard 開発用プラグインを追加したのち、開発環境の初期設定を行います。

4. SD カードからの起動

開発中の作業ディスクとして使う SD カードを作成します。

• 開発編

1. コンソールの利用

Teaboard 上のコンソールを Eclipse 上から使います。

2. “Hello, world” プロセスベース編

Eclipse 上で、プロセスベースのプログラムを新規に作成して、Teaboard 上に転送して実行します。

3. “Hello, world” T-Kernel ベース編

前項と同じことを今度は T-Kernel ベースで行います。

4. 実習用プログラムの実行

Teaboard 付属 CD-ROM に含まれる実習用プログラム (ルーレット、タイマ、簡易ウェブサーバ) を Eclipse 上でメイクして、Teaboard 上で実行します。

1 セットアップ編

1.1 Teaboard とパソコンの USB 接続

Teaboard とパソコンを USB で接続するために、Windows にデバイスドライバをインストールします。

† Teaboard とパソコンを接続する方法は、USB とシリアル (RS-232) のどちらも可能ですが、本書では USB の場合で説明します。

USB の方が一般に高速にファイル転送などを行うことができます。また、USB は給電も兼ねていますので、給電と通信を一本の USB ケーブルだけで行うことができます。一方、シリアルの場合は、Teaboard への給電は USB または AC アダプタから給電する必要があります。

以下に Windows XP にデバイスドライバをインストールする手順を示します。詳しくは、Teaboard 付属 CD-ROM 内の『取扱説明書』の「1.2.1 Windows ホストと USB 接続する場合」を参照してください。

† パソコンと Teaboard を USB 接続する場合、USB ハブを介さず USB ケーブルで直接接続することをおすすめします。USB ハブを介すると問題が生じる場合があります。

(1) ドライバのインストール

Windows XP SP3 の場合

1. 新しいハードウェアの検出ウィザード開始

Teaboard のディップスイッチ DSW2 の 1 番と 2 番 を両方とも ON にしてから、Teaboard と Windows パソコンを USB ケーブルで接続してください。

Windows が Teaboard を検出してウィザードが開始されます (図 1.1)。Windows Update には接続せずに、 をクリックします。

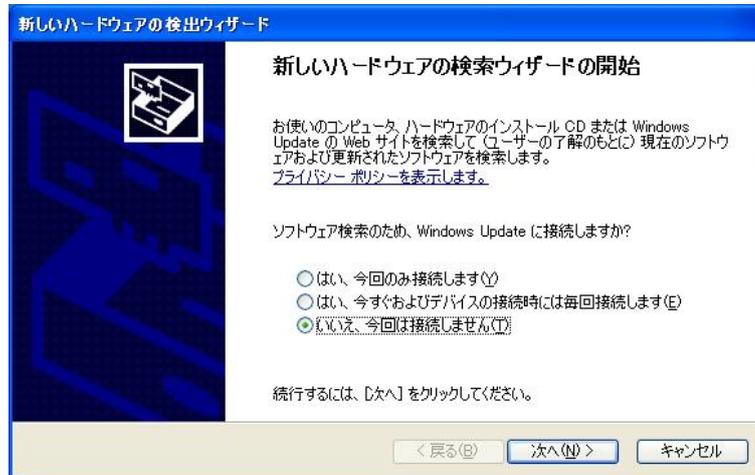


図 1.1 新しいハードウェアの検出ウィザード

2. ドライバ・セットアップファイルの選択

「一覧または特定の場所からインストールする」を選択して、**次へ(N) >** をクリックします。

「次の場所を含める」を選択して、ドライバ・セットアップファイル `teaboard.inf` のあるフォルダを指定し、**次へ(N) >** をクリックします。このセットアップファイルは、Teaboard 付属 CD-ROM 内の共通ソフトウェアのフォルダ `common%soft` にあります (図 1.2)。

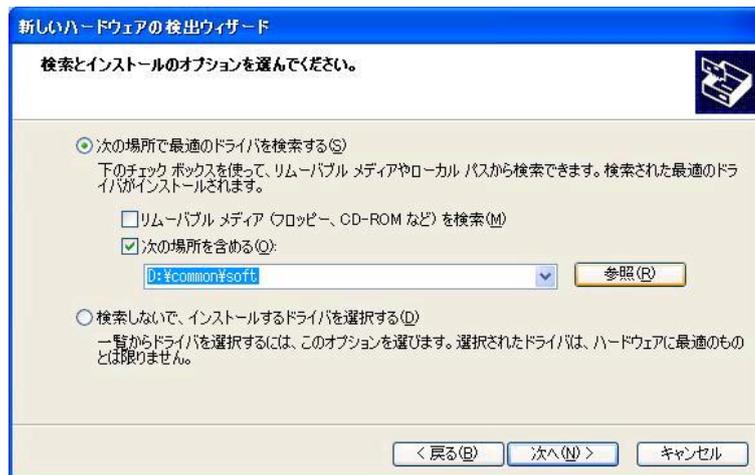


図 1.2 ドライバ・セットアップファイルの選択

3. ドライバのインストール

インストールが開始されますのでしばらくお待ち下さい。途中で「Windows ログテスト」に関する警告メッセージが出る場合もあ

りますが、正常ですので **続行 (C)** をクリックしてください。

しばらくするとインストール完了しますので、**完了** をクリックします。

Windows Vista, Windows 7 の場合

1. デバイスマネージャを起動します。

† デバイスマネージャは、「コントロールパネル」 「システムとセキュリティ」 「システム」 「デバイスマネージャ」にあります。

2. Teaboard のディップスイッチ DSW2 の 1 番と 2 番 を両方とも ON にしてから、Teaboard と Windows パソコンを USB ケーブルで接続します。

3. デバイスマネージャの「ほかのデバイス」の項目に「不明なデバイス」が追加されるので、これをダブルクリックしてプロパティを表示します。

† 不明なデバイスが複数表示される場合は、「詳細」タブをクリックし、プロパティのハードウェア ID の値に「USB\VID_17EE&PID_0001」が表示されるデバイスを選択してください。

4. 「ドライバーの更新 (U)」をクリックします。「どのような方法でドライバーソフトウェアを検索しますか?」に対して「コンピューターを参照してドライバーソフトウェアを検索します (R)」を選択します。

5. 「次の場所でドライバーソフトウェアを検索します」に、ドライバ・セットアップファイル teaboard_vista.inf のあるフォルダを指定し、**次へ (N)>** をクリックします。このセットアップファイルは、Teaboard 付属 CD-ROM 内の共通ソフトウェアのフォルダ common¥soft にあります。

6. Windows セキュリティの「ドライバーソフトウェアの発行元を検証できません」という警告が出ますので「このドライバーソフトウェアをインストールします (I)」をクリックします。

7. 「このデバイスのドライバーソフトウェアのインストールを完了しました」のメッセージを確認し、**閉じる (C)** をクリックします。

(2) COM ポート番号の確認

デバイスマネージャを起動して、「モデム」内の「PMC Teaboard」を選択します (図 1.3)。

- † デバイスマネージャは、Windows XP では「スタート」 「コントロールパネル」 「パフォーマンスとメンテナンス」 「システム」を開いて、「システムのプロパティ」の「ハードウェア」内にあります。

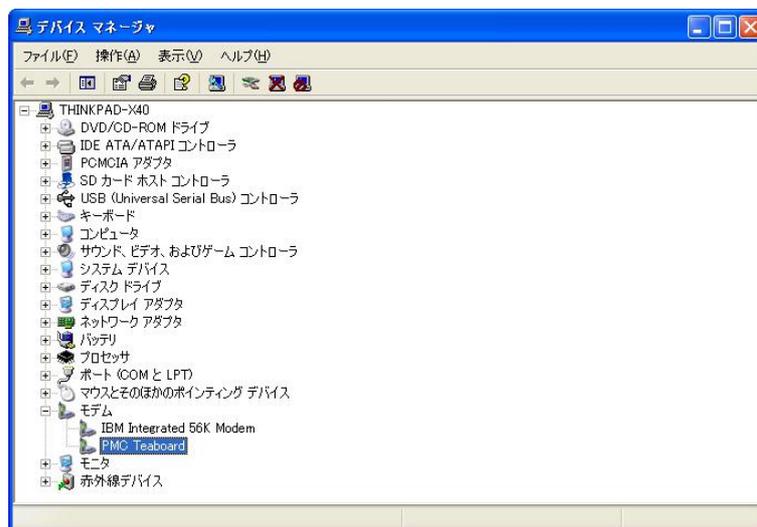


図 1.3 デバイスマネージャの起動

右クリックしてプロパティを開き、「モデム」タブを開いて、Teaboardに割り当てられた COM ポート番号を調べてください(図 1.4)。



図 1.4 COM ポート番号の確認

(3) 端末ソフトの起動

コンソール接続のための端末ソフトを起動します。以下では Windows 付属の「ハイパーターミナル」を使う例を示しますが、他の端末ソフト (TeraTerm Pro など) でも構いません。

† Windows Vista および Windows 7 にはハイパーターミナルが付属しません。端末ソフトをお持ちでない場合は「Tera Term Pro (バージョン 2.3)」を下記からダウンロードしてご利用ください。

<http://hp.vector.co.jp/authors/VA002416/>

† この節ではドライバのインストールのためにハイパーターミナルなどの端末ソフトを使いますが、次節以降で Eclipse の開発環境をインストールすれば、端末ソフトを使うかわりに Eclipse 上の gterm を使ってコンソール操作ができます。ただし、最初のセットアップ時には、接続がうまくいかない場合の原因切り分けのためにも、まずはハイパーターミナルなどの端末ソフトで接続を確認することをお勧めします。

ハイパーターミナル を利用する場合は、Windows のメニューの「スタート」「すべてのプログラム」「アクセサリ」「通信」「ハイパーターミナル」から起動します。名前とアイコンを選択する画面では、「Teaboard 接続テスト」などの適当な名前を入力して をクリックします。

† 「既定の Telnet プログラムにしますか?」と聞かれる場合がありますが、「はい」でも「いいえ」でもどちらでも構いません。また市外局番などを聞かれる場合もありますが、これも実際に電話をかけるわけではありませんので、適当に入力して構いません。

「接続の設定」では、Teaboard に割り当てられたシリアルポート ((4) で確認した「COM7」など) を選択して をクリックします。

続いてポートの設定では、次のように設定して をクリックします。

ビット/秒	115200 bps
データビット	8 ビット
パリティ	なし
ストップビット	1 ビット
フロー制御	ハードウェア

† TeraTerm Pro を使う場合、指定できる COM ポートの上限が足りない場合は、通常 C:\Program Files\TTERMPRO 内にある TERATERM.INI をメモ帳等で開いて、その中にある MaxComPort の設定を変更して下さい。

(4) 通信の確認

端末ソフト上で ↵ (Enter) キーを何回か押してみて、Teaboard 側からのメッセージ (T-Monitor のプロンプト TM> など) が表示されれば接続成功です (図 1.5)。



図 1.5 端末ソフト上に表示された T-Monitor のプロンプト

(5) ディップスイッチの設定

通信ソフトを終了した後に、Teaboard の USB ケーブルをパソコンから取り外して下さい。

Teaboard に電源が供給されていないことを確認した上で、Teaboard のディップスイッチ DSW2 の 1 番を (T-Kernel 起動) に戻して下さい。なおディップスイッチ DSW2 の 2 番は ON (USB 接続) のままにしてください。

(6) 次回以降の起動と終了

次回以降 Teaboard と端末ソフトを USB 経由で接続する際は、必ず Teaboard が起動した状態で、端末ソフトを起動したり終了したりするようにしてください。端末ソフトが起動した状態で、後から Teaboard 側を起動しても、接続できません。

特に端末ソフトが起動した状態で Teaboard 側に対してリセットをかけると、USB 接続が切断されてしまいます。このような場合は、先に端末ソフト側を終了してから Teaboard をリセットして、Teaboard の再起動後に端末ソフトを再起動するようにしてください。

ただしリセットに関しては、最新版のソフトウェアでは通信が切断されないように改善されましたので、端末ソフト側をいったん終了させる必要はありません。ただし Teaboard 側の T-Monitor/T-Kernel が最新版であることと、パソコン側の端末ソフトが最新版の te_vcom, gterm であることが必要です。古いバージョンをご利用の場合はバージョンアップをお願い申し上げます。

ただしこの改善はすべての端末ソフトに対応しているわけではありません。最新版の te_vcom, gterm については対応しています。

1.2 Cygwin のインストール

Windows パソコン上で開発環境を使うには、まず Cygwin をインストールする必要があります。

詳細については Teaboard 付属 CD-ROM 内の『Cygwin インストール方法説明書』をあわせてご参照ください。

† Windows のログイン時のアカウントは、管理者権限としてください。制限されたアカウントの場合、ソフトウェアのインストールがうまくいきません。また Windows のログイン時のアカウント名は、半角英数字のみとしてください。アカウント名に全角文字や半角空白などが入ると、Cygwin で不都合が生じる場合があります。

(1) Cygwin パッケージの展開

Teaboard 付属 CD-ROM 内の共通ソフトウェアのフォルダ (common¥soft) にある Cygwin システムパッケージ「cygwin.1.5.25-15.zip」を、いったんデスクトップなどに保存した上で、Windows のエクスプローラでマウスの右ボタンをクリックして「すべて展開 (A)...」を実行してください。展開先はハードディスクのどこでも構いません。

† ファイル名の中のバージョン番号は上記と異なる場合があります。

† Teaboard 付属 CD-ROM 内の Cygwin パッケージのかわりに、最新版の Cygwin をウェブサイト (<http://www.cygwin.com/>) からダウンロードすることも可能です。特に Windows Vista および Windows 7 の場合はダウンロードをお勧めします。詳細については Teaboard 付属 CD-ROM 内の『Cygwin インストール方法説明書』をご参照ください。

(2) インストーラの起動

展開したフォルダ内にあるインストーラ (setup.exe) をダブルクリックして起動します。

インストーラが起動したら をクリックします。

(3) インストールタイプの選択

「Install from Local Directory」を選択して をクリックします。

(4) インストールディレクトリの選択

Cygwin をインストールするディレクトリを選択します。デフォルトでは「C:¥cygwin」になっています。とくに支障がなければデフォルトのままにしておきます (図 1.6)。

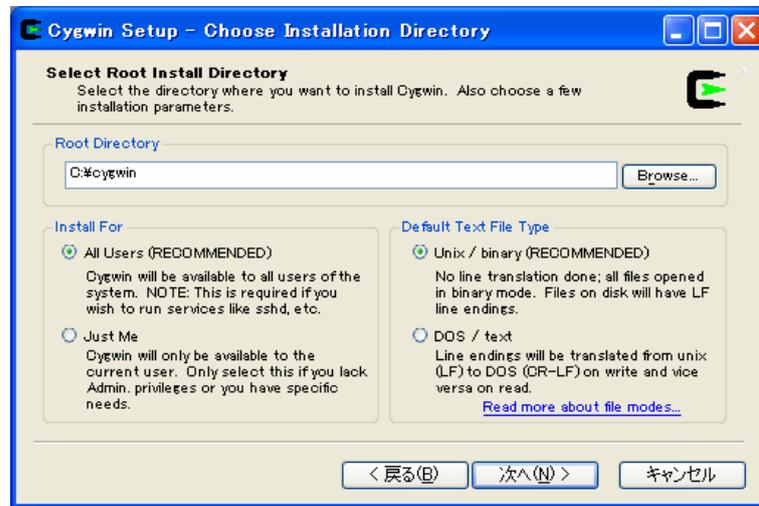


図 1.6 インストールディレクトリの選択

また「Install For」は「All Users」、「Default Text File Type」は「Unix」を選択してください。いずれもデフォルトのままです。

をクリックします。

(5) ローカルパッケージディレクトリの選択

(1) でインストールパッケージを展開したフォルダを指定して、 をクリックします。

(6) インストールするパッケージの選択

ファイルのチェックが行われた後、インストールパッケージの選択画面になります。「All」の右の「Default」をクリックして、すべて「Install」の状態にします (図 1.7)。

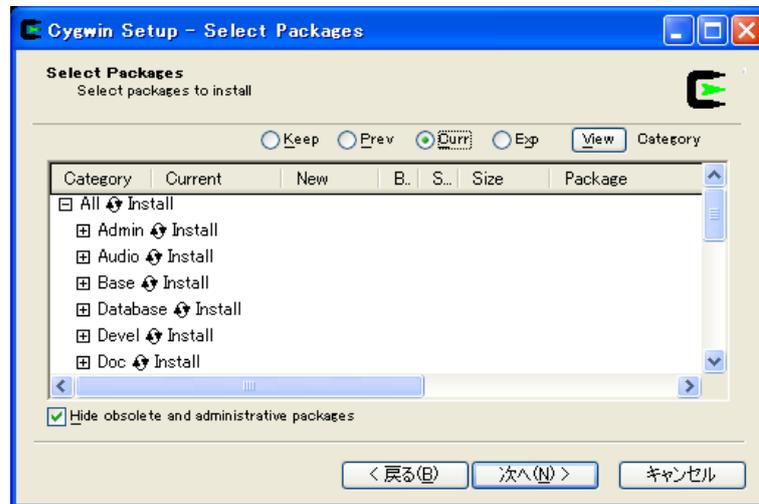


図 1.7 インストールするパッケージの選択

(7) インストール

をクリックすると、インストールが開始されます。時間がかかりますので、進捗が 100% になるまでしばらくお待ちください。

(8) アイコンの作成

「Create icon on Desktop」を選択して をクリックします。

をクリックすると、インストーラが終了します。

Cygwin をインストールしたら、次のように追加設定を行ってください。

(9) ファイル名の大文字と小文字の区別の設定

Cygwin はデフォルトではファイル名の大文字と小文字を区別しませんが、区別するように設定を変更する必要があります。

Cygwin 起動用バッチファイル「C:\cygwin\cygwin.bat」をテキストエディタで開き、「bash --login -i」という行の前に次の行を追加します。

```
set CYGWIN=nowinsymlinks check_case:strict
```

(10) home ディレクトリの初期化

デスクトップ上の「Cygwin」アイコンをダブルクリックして Cygwin を起動します。最初の起動時に home ディレクトリが初期化されます。

(11) gmake へのリンクの作成

gmake で make が起動するように、Cygwin 上で次のようなシンボリックリンクを作成します。

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

(12) Perl へのリンクの作成

Perl がパス名「/usr/local/bin/perl」で起動するように、Cygwin 上で次のようなシンボリックリンクを作成します。

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

終わったら Cygwin のウィンドウは `exit` コマンドで閉じて構いません。

```
$ exit↵
```

1.3 Eclipse と Teaboard 開発環境のインストール

統合開発環境の Eclipse をインストールして、その上に Teaboard 開発用のプラグインを追加します。

詳細については Teaboard 付属 CD-ROM 内の『Eclipse インストール方法説明書』と『Eclipse 用 T-Kernel 開発環境インストール方法説明書』をあわせてご参照ください。

(1) Java 実行環境のインストール

Eclipse の実行には Java 実行環境が必要です。まだインストールされていない場合、またはバージョンが古い場合は、

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

からダウンロードしてインストールしてください。

† 64 ビット版 OS の場合、32 ビット版の Java 実行環境が必要です。32 ビット版の Java 実行環境をインストールしてください。

(2) Eclipse システムパッケージの展開

Teaboard 付属 CD-ROM 内の共通ソフトウェアのフォルダ (common#soft) にある以下の 4 つのファイルをいったんデスクトップなどに保存した上で、Windows のエクスプローラでマウスの右ボタンをクリックして「すべて展開 (A)...」を実行してください。展開先は C ドライブの直下 (C:¥) を指定してください。

- Eclipse 本体
eclipse-platform-3.2.2-win32.zip
- Eclipse 本体用日本語化パック
NLpack1-eclipse-platform-3.2.1-win32.zip

- CDT プラグイン
org.eclipse.cdt-3.1.2-win32.x86.zip
- CDT 日本語化パック
CDT_NL_3.1.1.zip

† ファイル名の中のバージョン番号は上記と異なる場合があります。

† 展開先として C:¥ を指定すると、実際には C:¥eclipse というフォルダが作成され、その下に展開されます。

† CDT プラグインの展開時に、上書きされるかどうか 2 回聞かれますが、ライセンス表示の html ファイルですので、上書きで問題ありません。

(3) ショートカットの作成

「C:¥eclipse¥eclipse.exe」のショートカットをデスクトップ上に作成します。

(4) Eclipse の起動の確認

Eclipse のショートカットをダブルクリックして Eclipse を起動します。図 1.8 のような起動画面が表示され、それからワークスペースの選択ダイアログが出れば正常です。いったん をクリックして終了してください。



図 1.8 Eclipse の起動画面

【トラブルシューティング】「Java Runtime Environment がない」と表示される場合は、(1) の Java 実行環境のインストールをご確認ください。

(5) Teaboard 開発環境の Eclipse 用プラグインの展開

Teaboard 付属 CD-ROM 内の日本語版ソフトウェアのページ (jp¥soft) にある以下の 2 つのファイルをいったんデスクトップなどに保存した上で、Windows のエクスプローラでマウスの右ボタンをクリックして「すべて展開 (A)...」を実行してください。展開先は C:¥eclipse を指定してください。

プラットフォーム共通部分	com.t.engine4u.te.1.0.6.zip
各機種対応部分	com.t.engine4u.tl.tbmx1.1.0.2.zip

† zip アーカイブの展開は、Windows 標準の zip 展開機能で展開してください。他のソフトを使って展開すると、ソフトによっては正しく展開できない場合があります。

† ファイル名の中のバージョン番号は上記と異なる場合があります。

1.4 開発環境の初期設定

開発環境を起動して初期設定を行います。

(1) ターゲット側 (Teaboard) の起動

Teaboard を USB 経由でパソコンに接続します。USB から電源が供給され、Teaboard が起動します。

(2) Eclipse の起動

Eclipse のショートカットをダブルクリックして Eclipse を起動します。

(3) ワークスペースの選択

起動時にワークスペースの選択を聞かれます。ここではプロセスベースのプログラムを作成するための標準的なワークスペースである「C:\te\bapl」としてください。

† ワークスペースとは、プロジェクトを保管するフォルダのことです。通常、プロセスベースのプログラムを作成する場合は「C:\te\bapl」、T-Kernel ベースのプログラムを作成する場合は「C:\te\kapl」、デバイスドライバを作成する場合は「C:\te\driver」をワークスペースとするのが標準的ですが、これとは違う場所にワークスペースを作成しても構いません。

(4) ワークベンチにジャンプ

「ようこそ」ビューが表示されますので、ワークベンチをクリックします (図 1.9)。

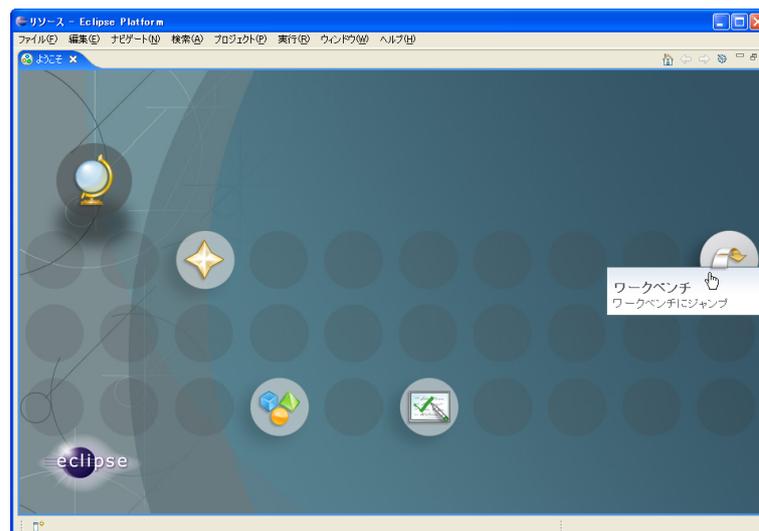


図 1.9 Eclipse の「ようこそ」ビュー

(5) T-Engine 開発パースペクティブを開く

ツールバーの「ウィンドウ」「パースペクティブを開く」「T-Engine 開発」を選択します (図 1.10)。

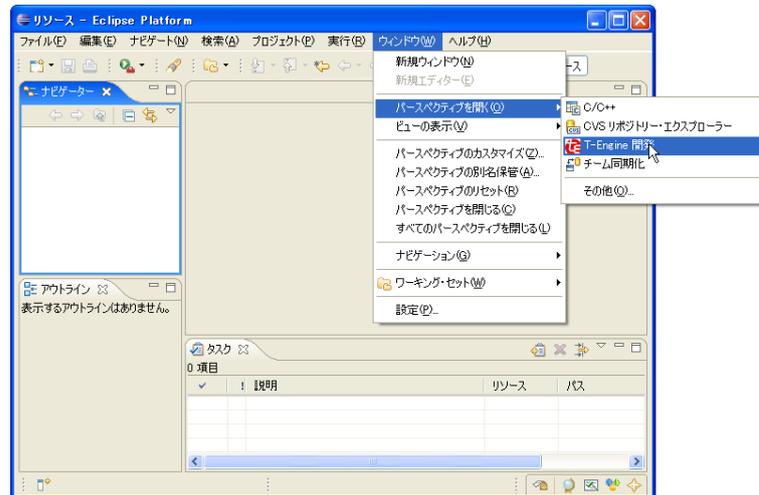


図 1.10 T-Engine 開発パースペクティブの選択

(6) T-Engine 開発環境の設定

ツールバーの「ウィンドウ」「設定」で設定ダイアログを開きます。設定ダイアログの左側の中から「T-Engine 開発環境」を選択して内容を確認します。

te_vcom の引数として、Teaboard の COM ポート番号から 1 を引いた値をデバイス名として指定します。例えば COM7 であれば「-B -1 /dev/ttyS6」を設定してください。

† -1 はマイナス・エルです。

(7) ワークスペースの設定

設定ダイアログの左側の中から「一般」の「ワークスペース」を選択して、次の設定を行います (図 1.11)。

- 「自動的にビルド」のチェックを外します。
- 「テキスト・ファイル・エンコード」の「その他」をチェックして、「EUC-JP」を選択します。
- 「新規テキスト・ファイルの行区切り文字」の「その他」をチェックして、「Unix」を選択します。

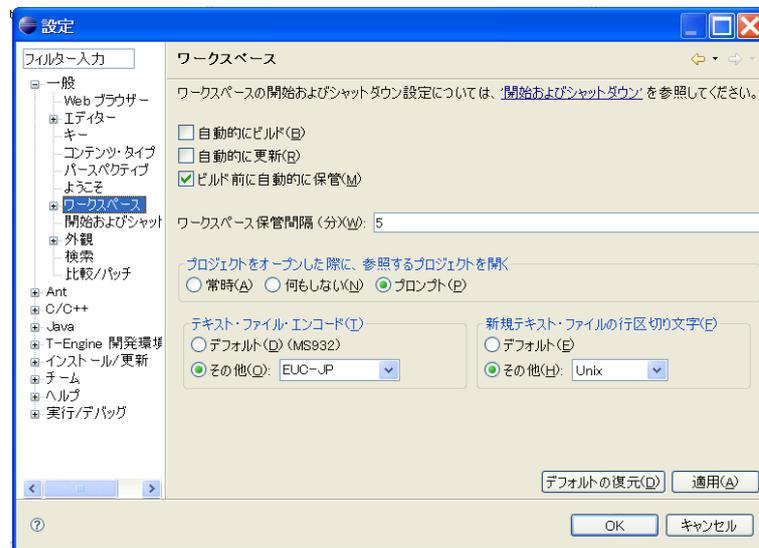


図 1.11 ワークスペースの設定

これらの環境設定はワークスペースごとに行う必要があります。プロセスベースのワークスペース「C:\te\bapp1」での設定が終わったら、ツールバーの「ファイル」 「ワークスペースの切り替え」でワークスペースを「C:\te\kapp1」(T-Kernel ベースのプログラム用のワークスペース) に切り替えて、同様の設定を行ってください。さらにワークスペースを「C:\te\driver」(デバイスドライバ用のワークスペース) に切り替えて、同様の設定を行ってください。

(8) ターゲット (Teaboard) 側との接続の確認

ターゲット (Teaboard) と通信を行うため、te_vcom (中継プログラム) と gterm(ターミナルエミュレータ) を起動します。ツールバーの外部ツールの をクリックし、「外部ツール (E)...」をクリックします (図 1.12)。

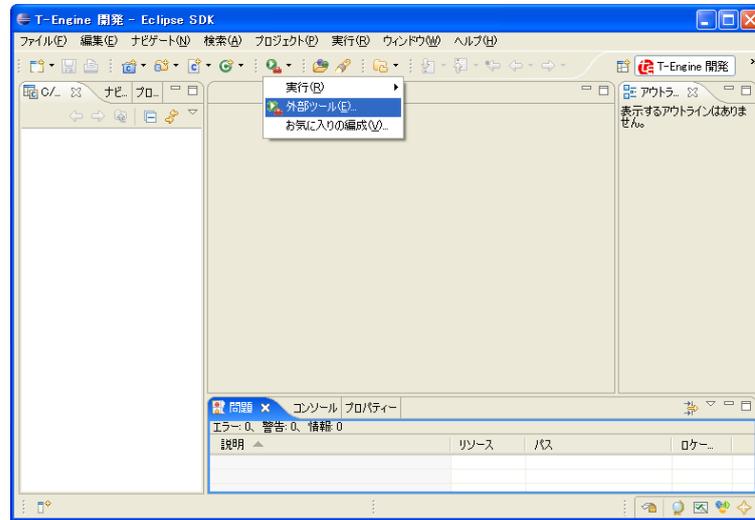


図 1.12 外部ツールの起動

外部ツールのダイアログが表示されたら、左側の「プログラム」の左の+をクリックすると「te_vcom」と「gterm」の設定が表示されます。設定内容は通常変更する必要はありません。

まずは「te_vcom」を選択し、**実行 (R)** をクリックしてください (図 1.13)。

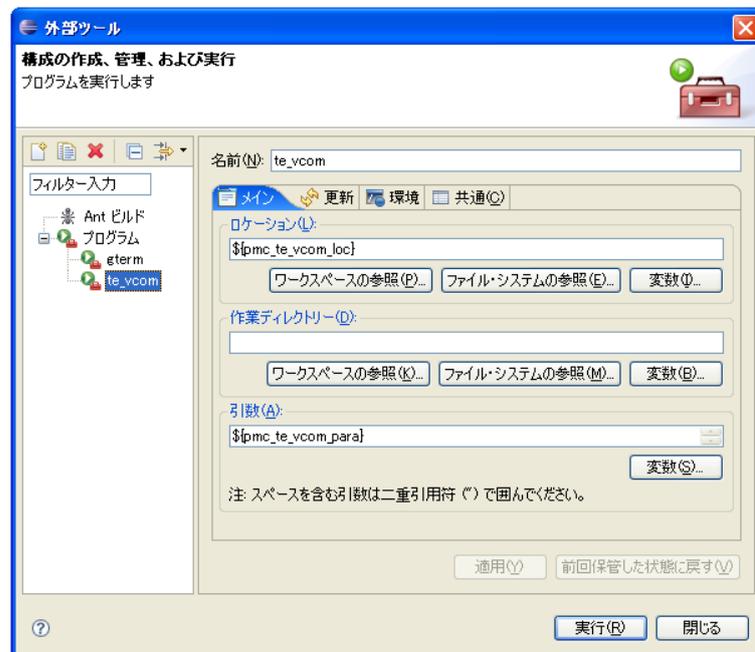


図 1.13 te_vcom の起動

続いて同様の手順により「gterm」を起動してください。コンソールに「gterm」の起動メッセージが表示されます。ターゲット側 (Teaboard) が起動した状態で、gterm のコンソール上で ↵ (Enter) キーを入力すると、ターゲット側から返されるプロンプト「[/SYS]#」が表示されます (図 1.14)。

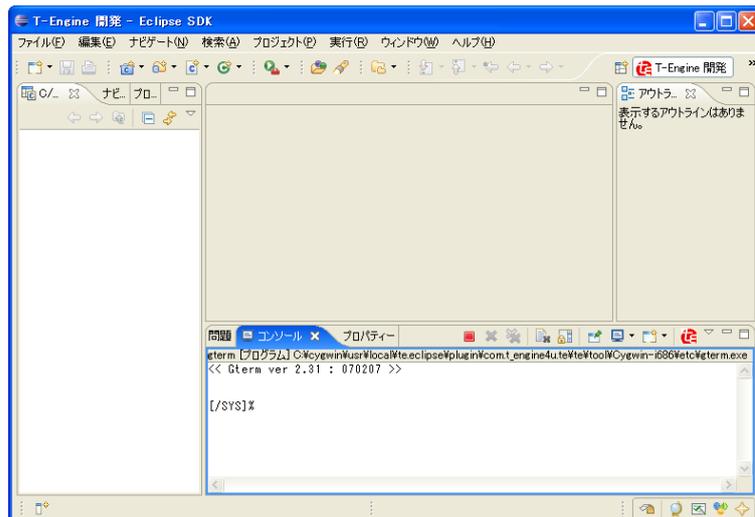


図 1.14 gterm 上に表示されたターゲット側プロンプト

これらの外部ツール te_vcom と gterm を一度起動すると、その後はツールバーの外部ツールのプルダウンリストにショートカットが表示されるようになりますので、次回からはこのショートカットを使って起動できます。

1.5 次回以降の起動

次回以降は、Teaboard と Eclipse を次のように起動してください。

(1) ターゲット側 (Teaboard) の起動

te_vcom や gterm などの端末ソフトが動作していない状態で、まず Teaboard を USB 経由でパソコンに接続します。USB から電源が供給され、Teaboard が起動します。

(2) Eclipse の起動

Eclipse を起動します。

最初にワークスペースの選択を聞かれます。プロセスベースの場合は「C:%te%bappl」、T-Kernel ベースの場合は「C:%te%kappl」、デバイスドライバの場合は「C:%te%driver」を選択します。

(3) te_vcom と gterm の起動

ツールバーの外部ツールの をクリックして、te_vcom と gterm を起動してください。

Eclipse を起動した直後や、ワークスペースを切り替えた直後は、te_vcom や gterm が起動されていません。毎回起動する必要がありますので、注意してください。

(4) CLI のプロンプトの確認

Eclipse のコンソールウィンドウの gterm で、↵ (Enter) キーを何回か押すことにより、CLI のプロンプト「[/SYS]%'」が表示されることを確認します。

† Teaboard 側をリセットする場合の注意

最新版のソフトウェアでは問題ありませんが、古いバージョンのソフトウェアをご利用の場合、Teaboard 上のリセットボタンを押すなどして Teaboard 側を再起動した場合は、いったん USB の接続が切断されてしまいます。Teaboard 側をリセットする必要がある場合は、コンソールウィンドウの上辺にある終了ボタン (赤い四角のボタン) を使って、あらかじめ gterm と te_vcom を終了させてから、Teaboard をリセットするようにしてください。Teaboard が再起動した後に、te_vcom と gterm を再起動します。

1.6 SD カードからの起動

Teaboard にはフラッシュ ROM があり、その中に ROM ディスクとして T-Kernel や各種設定ファイル、ツール類などが格納されています。しかし ROM ディスクですので、通常はこの内容を書き替えることはできません。

† ただし T-Monitor の “WriteRda” コマンドを使えば書き替えられます。Teaboard 付属 CD-ROM 内の『実装仕様書』の「2. T-Monitor 実装仕様」をご覧ください。

そのため ROM ディスクの内容を SD カードにコピーして、SD カードから起動する方が、設定ファイルの変更などにも対応しやすく、何かと便利で。また作成したプログラム等も SD カード上に転送してから実行する形になります。

Teaboard 付属 CD-ROM 内の『取扱説明書』の 3.1 節および 3.2 節にその詳しい説明があります。

SD カード上に起動ディスクを作成する手順例を以下に示します。この作業は gterm と CLI を使って行います。

(1) SD カードの区画 (パーティション) の設定

新しい (内容を消去しても構わない) SD カードを用意してください。

Teaboard の SD カードスロットに挿入してから、hdpart コマンドを

使って区画を作成し、ブート区画を設定します。SD カード全体の物理デバイス名は「pcb」です。

```
[/SYS]% hdpart pcb ← — SD カード (pcb) の分割
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 ← — 区画削除
Delete PartNo (1-4) ? 1 ← — 最初の区画
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 ← — 区画作成
Create PartNo (1-4) ? 1 ← — 最初の区画
Size [GB/MB/KB,All] (<243MB) ? a ← — 区画サイズ (最大)
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 ← — ブート区画
Boot PartNo (1-4,Clear) ? 1 ← — 最初の区画
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 ← — 区画更新
** pcb: Updated Master Boot Block
[/SYS]%
```

(2) 第 1 区画にシステムディスクを作成

第 1 区画のフォーマットとシステムディスク作成 (ROM ディスクからのコピー) を行います。第 1 区画の論理デバイス名は「pcb0」です。

```
[/SYS]% format -b pcb0 system ← — 第 1 区画のフォーマット
Format pcb0 [STD] system
Logical Formatting...
Writing BootCode...
Disk Format Success.
[/SYS]% att pcb0 A ← — 第 1 区画の接続
pcb0 -> pcb0
[/SYS]% rcp -b -r /SYS /A/= ← — システムディスクの全コピー
[/SYS]% det pcb0 ← — 第 1 区画の切断
[/SYS]%
```

(3) システムディスクの起動確認

Teaboard のリセットボタンを押してシステムを再起動します。

† 最新版のソフトウェアでは問題ありませんが、古いバージョンのソフトウェアをご利用の場合、コンソールを USB 接続している場合はリセットによっていったん接続が切れるので、あらかじめ gterm と te_vcom を終了させてから、Teaboard をリセットし、Teaboard の再起動後に te_vcom と gterm を再起動するようにしてください。

リセット後、「df」コマンドで、システムディスク (/SYS) が SD カードの第 1 区画を表す pcb0 になっていることを確認します。もし ROM ディスクを表す rda になっている場合は、うまくいっていませんので、手順を再度見直してください。

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

以上で SD カード上に起動ディスクが作成されました。以降、この起動ディスクを Teaboard に挿入して起動すればこの起動ディスクから起動しますので、この起動ディスクを使って実習を行って下さい。

2 開発編

2.1 コンソールの利用

Teaboard は開発に便利な各種ツール類が付属しています。プログラムの転送や実行については、gterm のコンソールからコマンドを手で入力しなくても、Eclipse のダイアログからプログラムの転送や実行ができるようになっています。しかしそれ以外のさまざまな用途のために、必要に応じて gterm のコンソールからこれらのツール類をご利用ください。

CLI (Command Line Interpreter)

プロセススペースのコマンドラインインタプリタです。開発に必要な、特にファイル関連操作などの強力な機能を提供します。プロンプトは標準では「[/SYS]%」です。詳しくは Teaboard 付属 CD-ROM 内の『開発ツール説明書』の「3. CLI 説明書」をご覧ください。

CLI 上で動作するユーティリティ

hdpart (SD カードの区画作成)、format (SD カードのフォーマット) などの各種ユーティリティが CLI 上で使える外部コマンドとしてシステムディスク上に格納されています。詳しくは『開発ツール説明書』の「4. ユーティリティ」、「5. UNIX(ファイル)エミュレータコマンドツール」、「6. ネットワークユーティリティ」をご覧ください。

IMS (Initial Monitor System)

T-Kernel ベースのモニタです。プロンプトは標準では [IMS]% です。詳しくは『開発ツール説明書』の「2. IMS」をご覧ください。

T-Monitor

最も基本的なモニタです。ハードウェアレベルの操作などに威力を発揮します。プロンプトは TM> です。

T-Monitor のコマンド一覧は『T-Monitor 仕様書』の「3.3. コマンド一覧」をご覧ください。さらに Teaboard では T-Monitor にフラッシュ ROM 書込機能などが追加されています。追加されたコマンド一覧は『実装仕様書』の「2. T-Monitor 実装仕様」をご覧ください。

以下に CLI と T-Monitor の簡単な実習例を示します。

(1) CLI のプロンプト確認

Eclipse のコンソールウィンドウの gterm 上で ↵(Enter) キーを何回か押して、CLI のプロンプト [/SYS]% が表示されることを確認します。

```
[/SYS]%
```

(2) CLI コマンド一覧

CLI の “?” コマンドを入力して、CLI のコマンド一覧を表示させます。

```
[/SYS]% ?↵
                CLI コマンド一覧
att      det      eject   cd       ls       fs       ...(後略)
...
```

(3) CLI コマンド詳細

CLI の “?” コマンドを使って “ls” コマンドの詳細ヘルプを表示させます。

```
[/SYS]% ? ls↵
ls [-f] [-F] [-l|-t] [<パス名>..]
<パス名>のファイルの直下にあるファイル一覧を表示する
...(後略) ...
```

(4) ファイル一覧

CLI の “ls” コマンドを使ってファイルの一覧を表示させます。

```
[/SYS]% ls↵
SBOOT          KERNEL.SYS     SYSCONF       ... (後略) ...
```

(5) T-Monitor への移行

CLI の 「#」 コマンドを使って T-Monitor へ移行します。

```
[/SYS]% #↵
TM>
```

(6) T-Monitor コマンド一覧

T-Monitor の “?” コマンドを使って T-Monitor のコマンド一覧を表示させます。

```
TM> ?↵
--- Command List :  "? command" for details ---
DumpByte/Half/Word(D/DB/DH/DW) ... (後略) ...
```

(7) T-Monitor コマンド詳細

T-Monitor の “?” コマンドを使って “OH” コマンドの詳細ヘルプを表示させます。

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

(8) I/O 書き込み

T-Monitor の “OH” コマンドを使って 0x16100000 番地に値 0 を書き込みます。この番地は右の 7 セグメント LED に接続されており、この LED が全点灯します。7 セグメント LED の制御方法については Teaboard 付属 CD-ROM 内の『ハードウェア仕様書』の「2.8 7 セグメント LED」をご参照下さい。

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

右の 7 セグメント LED に「3」というパターンを表示させます。「3」を表示させるには、セグメント f (0x02), セグメント e (0x08), 小数点 (0x20) を消灯し、残りを点灯すればいいので、0x16100000 番地に値 0x02 + 0x08 + 0x20 を書き込みます。

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

(9) T-Monitor からの復帰

T-Monitor の “G” (Go) コマンドを使って CLI へ復帰します。

```
TM> g↵
[/SYS]%
```

2.2 プロセスベースと T-Kernel ベース

Teaboard 上のプログラムは、大きく分けると「プロセスベースのプログラム」と「T-Kernel ベースのプログラム」の二つに分類されます。(このほか「モニタベースのプログラム」もありますが、本書では扱いません)

プロセスベースのプログラム

T-Kernel Extension 上で動作するプログラムです。

一般のアプリケーションを作成するのに向いています。

メモリ保護が効くので、プロセスに不具合があってもシステム全体に悪影響を及ぼす可能性は少なくなります。

メモリ空間としては、一つのプロセスに対して一つの独立したローカル空間が割り当てられ、そのローカル空間上で動作します。ただしプロセス内にサブタスクを生成することができ、その場合は一つのプロセス内の複数のタスクが同じ空間を共有します。

プロセスの実行では、まず main() 関数が実行され、main() 関数が終了するとプロセスも終了します。

プロセスベースのプログラムから使える API については、Teaboard 付属 CD-ROM 内の以下のドキュメントをご参照ください。

- 『T-Kernel Extension 説明書』
プロセスやタスクなどの基本機能、ファイル、イベントなどの仕様書です。
- 『TCP/IP マネージャ説明書』
ネットワーク機能 (TCP/IP) の説明書です。
- 『ライブラリ説明書』
C 言語標準ライブラリなどの説明書です。fopen() などのファイル操作ライブラリ (stdio.h) も含んでいます。
- 『デバイスドライバ説明書』
デバイスドライバを呼び出す場合に必要となる仕様書です。

T-Kernel ベースのプログラム

T-Kernel の機能を直接使うプログラムです。

ハードウェア制御や割込みなどを扱うことができるため、デバイスドライバなどを作成するのに向いています。

メモリ保護は効きません。

メモリ空間は全体で一つの空間を共有します。システムの共有空間内に複数の T-Kernel ベースのプログラムがロードされ常駐する形です。

T-Kernel ベースのプログラムは lodspg コマンドで共有空間上にロードします。この時に main() 関数が呼ばれ、第一引数 ac は渡される引数の個数を示します。通常この中でタスクやハンドラの生成などの初期化処理を行います。

また unlspg コマンドでアンロードすることができます。この時にも main() 関数が呼ばれ、第一引数 ac はマイナスの値です。通常この中で、最初に生成したタスクやハンドラの削除などの後処理を行います。

T-Kernel ベースのプログラムから使える API については、Teaboard 付属 CD-ROM 内の以下のドキュメントをご参照ください。

- 『T-Kernel 仕様書』
タスクやセマフォ、割込みハンドラなどのリアルタイム OS としての基本機能 (T-Kernel/OS) と、デバイスドライバ管理などの機能 (T-Kernel/SM) などの仕様書です。
- 『ライブラリ説明書』
C 言語標準ライブラリなどの説明書です。

† ライブラリ説明書に書かれたライブラリは、一部にプロセススペース専用、T-Kernel ベース専用のものもありますが、それ以外はどちらからも利用可能です。

- 『デバイスドライバ説明書』

デバイスドライバを作成したりデバイスドライバを呼び出す場合に必要となる仕様書です。

† デバイスドライバを呼び出すのはプロセススペースでも T-Kernel ベースでもどちらでも可能ですが、デバイスドライバを作成できるのは T-Kernel ベースだけです。

2.3 “Hello, world” プロセススペース編

ここではまずは有名な「Hello, world」プログラムを、プロセススペースで作成してみましょう。詳細については『GNU 開発環境 (Eclipse 版) 説明書』をあわせてご参照ください。

(1) ワークスペースの指定

プロセススペースのプログラムですので、Eclipse 起動時にワークスペースとして「C:\te\bapl」を指定してください。

既に別のワークスペースで Eclipse が起動している場合は、ツールバーの「ファイル」 「ワークスペースの切り替え」でワークスペースを「C:\te\bapl」に切り替えてください。

† ワークスペースを切り替えると Eclipse がいったん終了して te_vcom と gterm も終了しますので、再度 te_vcom と gterm を順番に起動する必要があります。その上で Eclipse のコンソールウィンドウの gterm で、↵(Enter) キーを何回か押して、CLI のプロンプト「[/SYS]%'が表示されることを確認しておきます。

(2) プロジェクトの新規作成

テンプレートを使ってプロジェクトを新規作成します。

「C/C++ プロジェクト」ビュー内で右クリックして(またはツールバーの「ファイル」から)、「新規」 「T-Engine C/C++ プロジェクト」を選択して、新規プロジェクト作成ダイアログで以下のように入力します(図 2.1)。

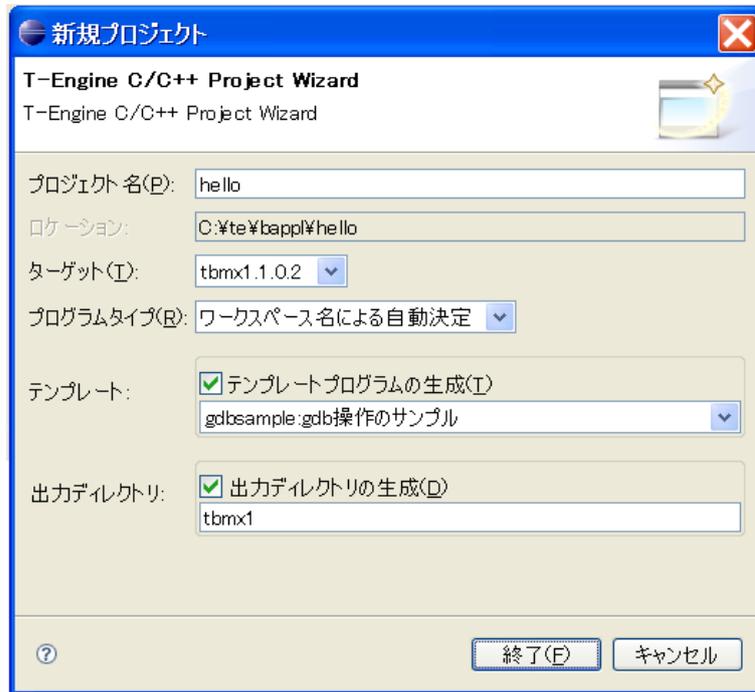


図 2.1 新規プロジェクト作成ダイアログ

- プロジェクト名:
プロジェクト名は自由につけることができますが、ここでは例えば「hello」とします。
- ターゲット:
複数の機種種のプラグインをインストールしている場合は、ここでターゲットの機種を選択します。
- プログラムタイプ:
「Process Base」を指定します。ワークスペースを「C:\te\bappl」としていれば、「ワークスペース名による自動決定」のままでも自動的にプロセスベースになります。
- テンプレート:
チェックを入れて、テンプレートをここでは「gdbsample:gdb 操作のサンプル」を選択します。
- 出力ディレクトリの生成:
チェックを入れます。

最後に 終了 (F) をクリックするとプロジェクトが自動生成されます。

(3) ソースの作成

「C/C++ プロジェクト」ビュー内に「hello」プロジェクトが生成されますので、ダブルクリックして開くとソースプログラムなどが参照できます。

ソースを次のように修正して、「Hello, world」を作成しましょう。

- hello/src/main.c :

たとえば以下のように修正してください。

```
/* Hello, world (プロセススペース) */
#include <basic.h>      /* 基本共通ヘッダ */
#include <stdio.h>     /* printf() など */
W main( W ac, TC *av[] )
{
    printf( "Hello, world\n" );
    return 0;
}
```

修正したら、ツールバーの「ファイル」 「保管」で修正を保存します。

† W や TC などは、T-Kernel 仕様書で定義された型です。ここではプロセスにコマンドライン引数を渡す意味で使っています。また return 0 としているのは、プロセス終了時に終了コードとして値 0 を返すためです。詳細は『T-Kernel Extension 説明書』などをご参照ください。

- hello/src/Makefile :

作成対象は「TARGET = hello」として、作成対象(メイクしてできる実行ファイル名)を hello に変更します。

ソースファイルは「SRC = main.c」とします。(sub.c は削除します)

修正したら、ツールバーの「ファイル」 「保管」で修正を保存します。

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

今回は不要なので削除します。右クリックメニューの中の「削除」で削除できます。

(4) メイク

メイクするターゲットとして「C/C++ プロジェクト」ビュー内の「hello/tbmx1/Makefile」を選択した上で、ツールバーの「プロジェクト」 「T-Engine Target の Make all」でメイクします。

メイクが成功すると、「hello/tbmx1/」の下に「hello」という名前で実行ファイルが生成されます(人が走っている図形で表示されます)。

【トラブルシューティング】 ファイルの転送と実行は CLI を使っていますので、CLI との通信ができないとうまくいきません。

このため Eclipse を起動したら、最初にツールバーの外部ツールから `te_vcom` と `gterm` を順番に起動しておく必要があります。その上で Eclipse のコンソールウィンドウの `gterm` で、`↵`(Enter) キーを何回か押して、CLI のプロンプト `[/SYS]%` が表示されることを確認しておきます。

【トラブルシューティング】 「Can't Create (-1966080)」と表示される場合：このエラーは、起動ディスクがリードオンリーで書き込みができないことを示しています。書き込み不可能なディスクから起動しているところのような現象になります。

2.4 「Hello, world」T-Kernel ベース編

今回は「Hello, world」プログラムを T-Kernel ベースで作成してみましよう。ロード時に「Hello, world」と表示し、アンロード時に「See you again」と表示するようにします。

(1) ワークスペースの指定

T-Kernel ベースのプログラムですので、Eclipse 起動時にワークスペースとして「`C:%te%kapp1`」を指定してください。

既に別のワークスペースで Eclipse が起動している場合は、ツールバーの「ファイル」→「ワークスペースの切り替え」でワークスペースを「`C:%te%kapp1`」に切り替えてください。

† ワークスペースを切り替えると Eclipse がいったん終了して `te_vcom` と `gterm` も終了しますので、再度 `te_vcom` と `gterm` を順番に起動する必要があります。その上で Eclipse のコンソールウィンドウの `gterm` で、`↵`(Enter) キーを何回か押して、CLI のプロンプト `[/SYS]%` が表示されることを確認しておきます。

(2) プロジェクトの新規作成

テンプレートを使ってプロジェクトを新規作成します。

「C/C++ プロジェクト」ビュー内で右クリックして(またはツールバーの「ファイル」から)、「新規」→「T-Engine C/C++ プロジェクト」を選択して、新規プロジェクトダイアログで以下のように入力します。

- プロジェクト名:
プロジェクト名は自由につけることができますが、ここでは例えば「hello2」とします。
- ターゲット:
複数の機種種のプラグインをインストールしている場合は、ここでターゲットの機種種を選択します。

- プログラムタイプ:
「T-Kernel Base」を指定します。ワークスペースを「C:¥te¥kapp1」としていれば、「ワークスペース名による自動決定」のままでも自動的に T-Kernel ベースになります。
- テンプレート:
チェックを入れて、テンプレートをここでは「sample:文字列出力サンプル」を選択します。
- 出力ディレクトリの生成:
チェックを入れます。

最後に 終了 (F) をクリックするとプロジェクトが自動生成されます。

(3) ソースの作成

「C/C++ プロジェクト」ビュー内に「hello2」プロジェクトが生成されますので、ダブルクリックして開くとソースプログラムなどが参照できます。

ソースを次のように修正して、「Hello, world」を作成しましょう。

- hello2/src/main.c :

たとえば以下のように修正してください。

```

/* Hello, world (T-Kernel ベース) */
#include <basic.h>          /* 基本共通ヘッダ */
#include <tk/tkernel.h>    /* T-Kernel ヘッダ */
#include <stdio.h>         /* printf() など */

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

```

修正したら、ツールバーの「ファイル」 「保管」で修正を保存します。

- hello2/src/Makefile :

作成対象は「TARGET = hello2」として、作成対象 (メイクしてできる実行ファイル名) を hello2 に変更します。

修正したら、ツールバーの「ファイル」「保管」で修正を保存します。

(4) メイク

メイクするターゲットとして「C/C++ プロジェクト」ビュー内の「hello2/tbmx1/Makefile」を選択した上で、ツールバーの「プロジェクト」「T-Engine Target の Make all」でメイクします。

メイクが成功すると、「hello2/tbmx1/」の下に「hello2」という名前で、リロケータブル形式の実行ファイルが生成されます(人が走っている図形では表示されませんが正常です)。このほか絶対アドレス形式の hello2.abs と、それを strip してサイズを小さくした hello2.trg も作成されます。

(5) 実行ファイルの転送と実行

転送する実行ファイルとして「C/C++ プロジェクト」ビュー内の「hello2/tbmx1/hello2」を選択した上で、右クリックメニューの「実行」「構成および実行」を選択します。

「構成および実行」ダイアログが表示されますので、「T-Engine アプリケーション」で右クリックして「新規」を選択します。「hello2」の転送や実行方法などの設定が自動設定されます。

実行 (R) をクリックすると、自動的に転送 (recv コマンド) と共有空間上にロード (lodspg コマンド) が行われ、「Hello, world」が表示されます。

(6) 共有空間の表示

Eclipse のコンソールウィンドウの gterm で、共有空間上にロードされているプログラムの一覧を「ref spg」コマンドで見て、新たに hello2 がロードされていることを確認します。

```
[/SYS]% ref spg↵
[ 1] - 0x8026e000 - 6  rsdrv
[ 2] - 0x80282000 - 7  netdrv
... (中略) ...
[ 5] - 0x80289000 - 7  hello2
[/SYS]%
```

† システムプログラム ID やアドレスは条件によって異なります。

(7) アンロード

前項で表示された hello2 の システムプログラム ID (上の例では 5) を指定して un1spg コマンドで hello2 をアンロードします。「See you again」と表示されます。

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

2.5 実習用プログラムの実行

Teaboard 付属 CD-ROM の「実習用教材集」のページには実習用プログラム集 exercise.zip が付属しています。Windows のエクスプローラ等で「C:¥te」上に展開してください。

なお、それぞれのプログラムの詳細につきましては、

TRONWARE VOL. 95 (パーソナルメディア刊)

に解説記事が掲載されておりますので、あわせてご参照いただければ幸いです。TRONWARE 誌につきましては、下記ページをご参照下さい:

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

2.5.1 ルーレット

T-Kernel ベースのプログラムで、7 セグメント LED を使ったルーレットです。

(1) ワークスペースの指定

T-Kernel ベースのプログラムですので、Eclipse 起動時にワークスペースとして C:¥te¥kapp1 を指定してください。

既に Eclipse が起動している場合は、ツールバーの「ファイル」「ワークスペースの切り替え」でワークスペースを C:¥te¥kapp1 に切り替えてください。

† ワークスペースを切り替えると Eclipse がいったん終了して te_vcom と gterm も終了しますので、再度 te_vcom と gterm を順番に起動する必要があります。その上で Eclipse のコンソールウィンドウの gterm で、↵(Enter) キーを何回か押して、CLI のプロンプト [/SYS]% が表示されることを確認しておきます。

(2) プロジェクトの新規作成

プロジェクトを新規作成します。今回は既にソースファイルは展開済みですので、テンプレートは使用しません。

「C/C++ プロジェクト」ビュー内で右クリックして(またはツールバーのファイルから)、「新規」「T-Engine C/C++ プロジェクト」を選択して、新規プロジェクトダイアログで以下のように入力します。

- プロジェクト名:
ここでは roulette とします。
- ターゲット:
複数の機種種のプラグインをインストールしている場合は、ここでターゲットの機種種を選択します。
- プログラムタイプ:
T-Kernel Base を指定します。ワークスペースを C:¥te¥kapp1 としていれば、「ワークスペース名による自動決定」のままでも自動的に T-Kernel ベースになります。
- テンプレート:
チェックを入れません。
- 出力ディレクトリの生成:
チェックを入れます。

最後に をクリックするとプロジェクトが自動生成されます。

「C/C++ プロジェクト」ビュー内に「roulette」プロジェクトが生成されますので、ダブルクリックして開くと中が参照できます。

(3) メイク

メイクするターゲットとして「C/C++ プロジェクト」ビュー内の「roulette/tbmx1/Makefile」を選択した上で、ツールバーの「プロジェクト」 「T-Engine Target の Make all」でメイクします。

メイクが成功すると、「roulette/tbmx1/」の下に「roulette」という名前で実行ファイルが生成されます (人が走っている図形では表示されませんが正常です)。

(4) 実行ファイルの転送と実行

転送する実行ファイルとして「C/C++ プロジェクト」ビュー内の「roulette/tbmx1/roulette」を選択した上で、右クリックメニューの「実行」 「構成および実行」を選択します。

「構成および実行」ダイアログが表示されますので、「T-Engine アプリケーション」で右クリックして「新規」を選択します。「roulette」の転送や実行方法などの設定が自動設定されます。

をクリックすると、自動的に転送 (recv コマンド) と共有空間上にロード (lodspg コマンド) が行われます。

7 セグメント LED の外周の 8 つのセグメントが順に光って回転します。右プッシュスイッチ SW3 を押すと回転と停止が切り替わります。左プッシュスイッチ SW2 を押すと終了します。

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

このプログラムは常駐しないように、意図的にエラーとして値 `-1` を返していますので、このメッセージは正常です。

なおこのプログラムは左右のプッシュスイッチに対する割込みを禁止状態とします。

2.5.2 タイマー

プロセススペースのプログラムで、指定された秒数だけカウントダウンして、ゼロになったらブザーを鳴します。デバイスドライバを使う例として、7 セグメント LED の表示は LED 制御ドライバ、ブザーを鳴すのは ブザー制御ドライバにそれぞれ任せています。

(1) ワークスペースの指定

LED 制御ドライバとブザー制御ドライバはデバイスドライバですので、Eclipse 起動時にワークスペースとして `C:%te%driver` を指定してください。

既に Eclipse が起動している場合は、ツールバーの「ファイル」「ワークスペースの切り替え」でワークスペースを `C:%te%driver` に切り替えてください。

† ワークスペースを切り替えると Eclipse がいったん終了して `te_vcom` と `gterm` も終了しますので、再度 `te_vcom` と `gterm` を順番に起動する必要があります。その上で Eclipse のコンソールウィンドウの `gterm` で、`↵`(Enter) キーを何回か押して、CLI のプロンプト `[/SYS]%` が表示されることを確認しておきます。

(2) プロジェクトの新規作成

プロジェクトを新規作成します。今回は既にソースファイルは展開済みですので、テンプレートは使用しません。

「C/C++ プロジェクト」ビュー内で右クリックして(またはツールバーのファイルから)、「新規」「T-Engine C/C++ プロジェクト」を選択して、新規プロジェクトダイアログで以下のように入力します。

- プロジェクト名:
ここでは `led` とします。
- ターゲット:
複数の機種種のプラグインをインストールしている場合は、ここでターゲットの機種種を選択します。

- プログラムタイプ:

Driver を指定します。ワークスペースを `C:%te%driver` としていれば、「ワークスペース名による自動決定」のままでも自動的にデバイスドライバになります。

- テンプレート:

チェックを入れません。

- 出力ディレクトリの生成:

チェックを入れます。

最後に **終了 (F)** をクリックするとプロジェクトが自動生成されます。

「C/C++ プロジェクト」ビュー内に「led」プロジェクトが生成されますので、ダブルクリックして開くと中が参照できます。

(3) メイク

メイクするターゲットとして「C/C++ プロジェクト」ビュー内の「led/tbmx1/Makefile」を選択した上で、ツールバーの「プロジェクト」→「T-Engine Target の Make all」でメイクします。

メイクが成功すると、「led/tbmx1/」の下に「led」という名前で実行ファイルが生成されます (人が走っている図形では表示されませんが正常です)。

(4) 実行ファイルの転送と実行

転送する実行ファイルとして「C/C++ プロジェクト」ビュー内の「led/tbmx1/led」を選択した上で、右クリックメニューの「実行」→「構成および実行」を選択します。

「構成および実行」ダイアログが表示されますので、「T-Engine アプリケーション」で右クリックして「新規」を選択します。「led」の転送や実行方法などの設定が自動設定されます。**実行** をクリックすると、自動的に転送 (recv コマンド) と共有空間上にロード (lodspg コマンド) が行われます。

† ロードしたデバイスドライバは、他のプログラムから呼ばれるまでは特に使われませんので、この段階ではまだ動作は見えません。

(5) 共有空間の表示

Eclipse のコンソールウィンドウの gterm で、共有空間上にロードされているプログラムの一覧を「ref spg」コマンドで見て、新たに「led」がロードされていることを確認します。

```

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

```

† システムプログラム ID やアドレスは条件によって異なります。

(6) ブザー制御ドライバのプロジェクト作成、メイク、実行

同様にブザー制御ドライバ「buzzer」についても、新規プロジェクト作成、メイク、実行を行います。

「ref spg」コマンドで「led」と「buzzer」が両方ロードされていることを確認してください。

(7) ワークスペースの切替

最後にプロセススペースのプログラム「timer」を作成して、LED 制御ドライバとブザー制御ドライバを呼び出します。

プロセススペースのプログラムですので、ツールバーの「ファイル」「ワークスペースの切り替え」でワークスペースを「C:¥te¥bappl」に切り替えてください。

† ワークスペースを切り替えると Eclipse がいったん終了して te_vcom と gterm も終了しますので、再度 te_vcom と gterm を順番に起動する必要があります。その上で Eclipse のコンソールウィンドウの gterm で、↵(Enter) キーを何回か押して、CLI のプロンプト [/SYS]% が表示されることを確認しておきます。

(8) プロジェクトの新規作成

プロジェクトを新規作成します。今回は既にソースファイルは展開済みですので、テンプレートは使用しません。

「C/C++ プロジェクト」ビュー内で右クリックして(またはツールバーのファイルから)、「新規」「T-Engine C/C++ プロジェクト」を選択して、新規プロジェクトダイアログで以下のように入力します。

- プロジェクト名:
ここでは timer とします。
- ターゲット:
複数の機種種のプラグインをインストールしている場合は、ここでターゲットの機種種を選択します。

- プログラムタイプ:

Process Base を指定します。ワークスペースを C:\te\bappl としていれば、「ワークスペース名による自動決定」のままでも自動的にプロセスベースになります。

- テンプレート:

チェックを入れません。

- 出力ディレクトリの生成:

チェックを入れます。

最後に **終了 (F)** をクリックするとプロジェクトが自動生成されます。

「C/C++ プロジェクト」ビュー内に「timer」プロジェクトが生成されますので、ダブルクリックして開くと中が参照できます。

(9) メイク

メイクするターゲットとして「C/C++ プロジェクト」ビュー内の「timer/tbmx1/Makefile」を選択した上で、ツールバーの「プロジェクト」 「T-Engine Target の Make all」でメイクします。

メイクが成功すると、「timer/tbmx1/」の下に「timer」という名前で実行ファイルが生成されます (人が走っている図形で表示されます)。

(10) 実行ファイルの転送と実行

転送する実行ファイルとして「C/C++ プロジェクト」ビュー内の「timer/tbmx1/timer」を選択した上で、右クリックメニューの「実行」 「構成および実行」を選択します。

「構成および実行」ダイアログが表示されますので、「T-Engine アプリケーション」で右クリックして「新規」を選択します。「timer」の転送や実行方法などの設定が自動設定されます。

実行 (R) をクリックすると、自動的に転送 (recv コマンド) と実行が行われます。

60 秒間のカウントダウンを行い、ゼロになったらブザーが鳴ります。

† ブザーについて:

- Teaboard2/ARM920-MX1 では、ハードウェアタイマ Timer2 をブザーに接続することで音を鳴らしています。
- Teaboard/ARM920-MX1 (初代) では、GIPO D pin 31 端子に H にすることでブザーを鳴らしています。ブザー ON の状態は最低でも 100 ミリ秒以上維持してください。

詳しくは『ハードウェア仕様書』の「2.18 ブザー」をご参照下さい。

2.5.3 簡易ウェブサーバ

簡単なウェブサーバです。Teaboard 上のファイルをネットワーク経由で配信して、他のマシン上のブラウザから見ることができます。

プロセススペースのプログラムです。ネットワーク通信には TCP/IP 機能を使っています。

(1) ワークスペースの指定

プロセススペースのプログラムですので、Eclipse 起動時にワークスペースとして「C:%te%bapp1」を指定してください。

既に別のワークスペースで Eclipse が起動している場合は、ツールバーの「ファイル」→「ワークスペースの切り替え」でワークスペースを「C:%te%bapp1」に切り替えてください。

† ワークスペースを切り替えると Eclipse がいったん終了して te_vcom と gterm も終了しますので、再度 te_vcom と gterm を順番に起動する必要があります。その上で Eclipse のコンソールウィンドウの gterm で、↵(Enter) キーを何回か押して、CLI のプロンプト [/SYS]% が表示されることを確認しておきます。

(2) ネットワークの設定

Eclipse のコンソールウィンドウの gterm 上で ↵(Enter) キーを何回か押して、CLI のプロンプト [/SYS]% が表示されることを確認したのち、「netconf」コマンドを使ってネットワークの設定を行います。

ご使用のネットワーク環境が DHCP による IP アドレスの自動割当てを行う場合には標準設定の 0.0.0.0 のままで問題ありません。そうでない場合には、「netconf c」コマンドで Teaboard に IP アドレスを割り当てて下さい。

```
[/SYS]% netconf c ↵
hostname = ? teaboard↵          — 自ホスト名
host ip   = 0.0.0.0 ? 192.168.0.70↵ — 自 IP アドレス
dns1name  = ? ↵                  — DNS サーバ 1
dns1 ip   = 0.0.0.0 ? ↵
dns2name  = ? ↵                  — DNS サーバ 2
dns2 ip   = 0.0.0.0 ? ↵
domain    = ? ↵                  — ドメイン名
gateway ip = 0.0.0.0 ? ↵          — ゲートウェイ
subnetmask = 255.255.255.0 ? ↵    — サブネットマスク
wlan      = none (n/a/i)? ↵       — 有線 LAN
[/SYS]%
```

(3) ネットワークの動作確認

Eclipse のコンソールウィンドウの gterm 上で、ping コマンドでネットワークの接続を確認します。

```
[/SYS]% ping localhost ← — 自分への ping
localhost is alive <192.168.0.70> : 0 ms
[/SYS]% ping 192.168.0.1 ← — 他のマシンへの ping
192.168.0.1 is alive <192.168.0.1> : 0 ms
```

† この例では Teaboard の IP アドレスは 192.168.0.70、他のマシンの IP アドレスは 192.168.0.1 ですが、それぞれご使用のネットワーク環境に合わせて適宜読み替えて下さい。

(4) 画像ファイルの転送

ウェブサーバで配信するコンテンツとなる画像ファイルを、ターゲット側に転送しておきます。

Eclipse のコンソールウィンドウの gterm 上で、次のように recv コマンドを使って、開発環境側のパソコン上にある画像ファイル(対応している形式は拡張子が .jpg または .png であるもの)を転送します。

例えば「C:%tmp%abc.jpg」を転送する場合は次のようにします。

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

同様に画像ファイルを数枚転送しておきます。

(5) プロジェクトの新規作成

プロジェクトを新規作成します。今回は既にソースファイルは展開済みですので、テンプレートは使用しません。

「C/C++ プロジェクト」ビュー内で右クリックして(またはツールバーのファイルから)、「新規」 「T-Engine C/C++ プロジェクト」を選択して、新規プロジェクトダイアログで以下のように入力します。

- プロジェクト名:
ここでは webserv とします。
- ターゲット:
複数の機種のプラグインをインストールしている場合は、ここでターゲットの機種を選択します。
- プログラムタイプ:
Process Base を指定します。ワークスペースを C:%te%bapl としていれば、「ワークスペース名による自動決定」のままでも自動的にプロセスベースになります。

- テンプレート:
チェックを入れません。
- 出力ディレクトリの生成:
チェックを入れます。

最後に **終了 (F)** をクリックするとプロジェクトが自動生成されます。
「C/C++ プロジェクト」ビュー内に「webserv」プロジェクトが生成されますので、ダブルクリックして開くと中が参照できます。

(6) メイク

メイクするターゲットとして「C/C++ プロジェクト」ビュー内の「webserv/tbmx1/Makefile」を選択した上で、ツールバーの「プロジェクト」 「T-Engine Target の Make all」でメイクします。
メイクが成功すると、「webserv/tbmx1/」の下に「webserv」という名前で実行ファイルが生成されます (人が走っている図形で表示されます)。

(7) 実行ファイルの転送と実行

転送する実行ファイルとして「C/C++ プロジェクト」ビュー内の「webserv/tbmx1/webserv」を選択した上で、右クリックメニューの「実行」 「構成および実行」を選択します。
「構成および実行」ダイアログが表示されますので、「T-Engine アプリケーション」で右クリックして「新規」を選択します。「webserv」の転送や実行方法などの設定が自動設定されます。

実行 (R) をクリックすると、自動的に転送 (recv コマンド) と実行が行われます。

(8) ブラウザからの閲覧

ネットワーク上の別のパソコンのブラウザから、Teaboard の IP アドレスを指定して、ブラウザ上に画像一覧が配信されることを確認します。例えば Teaboard の IP アドレスが 192.168.0.70 であればブラウザから次のように URL を入力して下さい:

「http://192.168.0.70/index.html」

† index.html を指定すると、画像ファイルの一覧画面が自動作成されて配信されます。

(9) プロセス終了

最後にこのウェブサーバを終了させたい場合は、コンソールから ^C (Ctrl-C) を送ることでプロセスを終了させてください。Eclipse のコンソールウィンドウの上辺の右端付近の TE ロゴマークのボタンを使って ^C を送信することができます (図 2.4)。



図 2.4 Eclipse での ^C の送信

索引

	C		
CLI	26	
Cygwin	12	
	D		
df コマンド	25	
	E		
Eclipse	15	
	F		
format コマンド	24	
	H		
hdpart コマンド	24	
	I		
IMS	26	
	J		
Java 実行環境	15	
	L		
LED	37	
lodspg コマンド	29	
ls コマンド	27	
	N		
netconf コマンド	43	
	P		
Perl	15	
ping コマンド	44	
	R		
recv コマンド	44	
	T		
T-Kernel ベース	29	
T-Monitor	26	
TCP/IP	43	
			TeraTerm Pro 10
	U		
unlspg コマンド	29	
	W		
Windows 7	8	
Windows Vista	8	
Windows XP SP3	6	
	あ		
ウェブサーバ	43	
	か		
コンソール	26	
	た		
タイマー	39	
端末ソフト	10	
デバイスマネージャ	9	
	は		
ハイパーターミナル	10	
ブザー	42	
プロジェクト	30	
プロセススペース	28	
	ま		
メモリ保護	28	
	ら		
リセット	23	
ルーレット	37	
	わ		
ワークスペース	18	

はじめてみよう Teaboard(Eclipse 環境編)
パーソナルメディア株式会社
〒141-0031 東京都品川区西五反田 1-29-1 コイズミビル
Tel: 03-5759-8305 Fax: 03-5759-8306
Web: <http://www.t-engine4u.com/>
E-Mail: te-sales@personal-media.co.jp
Copyright © 2005-2010 by Personal Media Corporation
