

# T-Kernel 2.0 リファレンスキット 取扱説明書

---

1.00.02  
2013年11月

## 目次

1. はじめに	1
1.1 製品構成	1
1.2 ハードウェア取り扱い方法	3
1.3 T-Kernel 2.0 リファレンスキットの起動方法	5
1.4 USB での接続	5
1.4.1 はじめて Windows ホストに USB 接続する場合	5
1.4.2 接続方法(シリアル/USB)の変更	6
1.4.3 USB デバッグポートを使用する際の注意事項	7
1.5 起動ディスクの作成	7
1.5.1 基本機能	7
1.5.2 PMC T-Shell の追加	9
1.5.3 PMC T-Shell インストール後の動作確認	11
1.6 フラッシュメモリの再書き込み	11
2. ソフトウェア構成	13
2.1 全体構成	13
2.2 ファイル構成	15
2.3 システムの起動手順	17
3. システムコンフィグレーション	19
3.1 GUI 機能	19
3.2 ユーザ環境	20
3.3 画面カスタマイズ	20
3.4 フォント	20
3.5 ネットワーク	22
4. ソフトウェア開発方法	25
4.1 開発環境のインストール	25
4.1.1 Linux 環境で開発を行う場合	25
4.1.2 Windows 上のコマンドライン環境で開発を行う場合	25
4.1.3 Windows 上の GUI 環境で開発を行う場合	25
4.1.4 PMC T-Shell 開発環境のインストール	26
4.2 開発対象ソフトウェアの分類	27
4.2.1 モニタベースのソフトウェア	27
4.2.2 T-Kernel ベースのソフトウェア	28
4.2.3 プロセスベースのソフトウェア	28
4.3 プログラム開発方法	28
4.4 デバッグモード	29
4.5 ROM ディスクの更新	29
4.6 T-Kernel 2.0 対応について	30

【修正履歴】

Version 1.00.00

- ・新規

Version 1.00.01

- ・ Windows のデバイスドライバ URL の誤植を修正。
- ・ Windows のデバイスドライバのインストール方法を Windows XP、Vista, 7 で共通化。

Version 1.00.02

- ・ Windows のUSBデバイスドライバを更新。

## 1. はじめに

### 1.1 製品構成

T-Kernel 2.0 リファレンスキット には以下の内容が含まれています。

#### ○ T-Kernel 2.0 リファレンスキット ハードウェア

詳細は「T-Engine Reference Board Hardware Manual」を参照してください。

#### ○ 製品マニュアル

T-Kernel 2.0 リファレンスキット取扱説明書(PDF) (本書)  
本製品の内容および操作方法に関する説明書です。

PMC T-Kernel Extension 説明書(PDF)  
本製品に含まれる PMC T-Kernel Extension に関する説明書です。

PMC T-Shell 説明書 (HTML形式)  
PMC T-Shell の API 説明書です。

PMC T-Shell プログラミング解説書 (PDF形式)  
PMC T-Shell のプログラミング解説書です。

ライブラリ説明書(PDF)  
本製品に含まれるC言語ライブラリに関する説明書です。

T-Engine 開発キットデバイスドライバ共通説明書(PDF)  
本製品や、T-Engine 共通に含まれるデバイスドライバに関する説明書です。

ネットワークデバイスドライバ説明書(PDF)  
本製品に含まれるネットワークドライバに関する説明書です。

USBマネージャ取扱説明書(PDF)  
本製品に含まれる USBマネージャに関する説明書です。

開発ツール説明書(PDF)  
本製品に含まれる PMC T-Kernel および PMC T-Kernel Extension 上で動作する開発ツールの説明書です。

T-Kernel 2.0 リファレンスキット GNU開発環境(Linux版)説明書(PDF)  
本製品に含まれる Linux 環境で動作する GNU 開発環境のインストール方法および操作方法に関する説明書です。

T-Kernel 2.0 リファレンスキット GNU開発環境(Windows版)説明書(PDF)  
本製品に含まれる Windows上の Cygwin 環境で動作する GNU 開発環境の操作方法に関する説明書です。

**Cygwin インストール方法説明書 (PDF)**

本製品に含まれる Cygwin 環境上で動作するプログラムを使用する場合に必要な Cygwin のインストール方法に関する説明書です。

**Cygwin 用 T-Kernel 2.0 リファレンスキット 開発環境インストール方法説明書 (PDF)**

本製品に含まれる T-Kernel 2.0 リファレンスキット GNU開発環境(Windows版)のインストール方法に関する説明書です。

**Eclipse インストール方法説明書 (PDF)**

本製品に含まれる GNU開発環境(Eclipse版)を使用する場合に必要な Eclipse のインストール方法に関する説明書です。

**Eclipse 用 T-Kernel開発環境インストール方法説明書 (PDF)**

本製品に含まれる GNU開発環境(Eclipse版)のインストール方法に関する説明書です。

**GNU開発環境(Eclipse版)説明書 (PDF)**

本製品に含まれる Windows上の GUI で動作する Eclipse版開発環境の操作方法に関する説明書です。

**○ 仕様書****T-Monitor 仕様書 (PDF)**

T-Monitor の公式仕様書です。

**T-Kernel 2.0 仕様書 (PDF)**

T-Kernel 2.0 の公式仕様書です。

**実装仕様書 (PDF)**

T-Monitor/T-Kernel そして、デバイスドライバの T-Kernel 2.0 リファレンスキットの実装に依存した部分の詳細な仕様書です。

**○ ソフトウェア****ターゲット側**

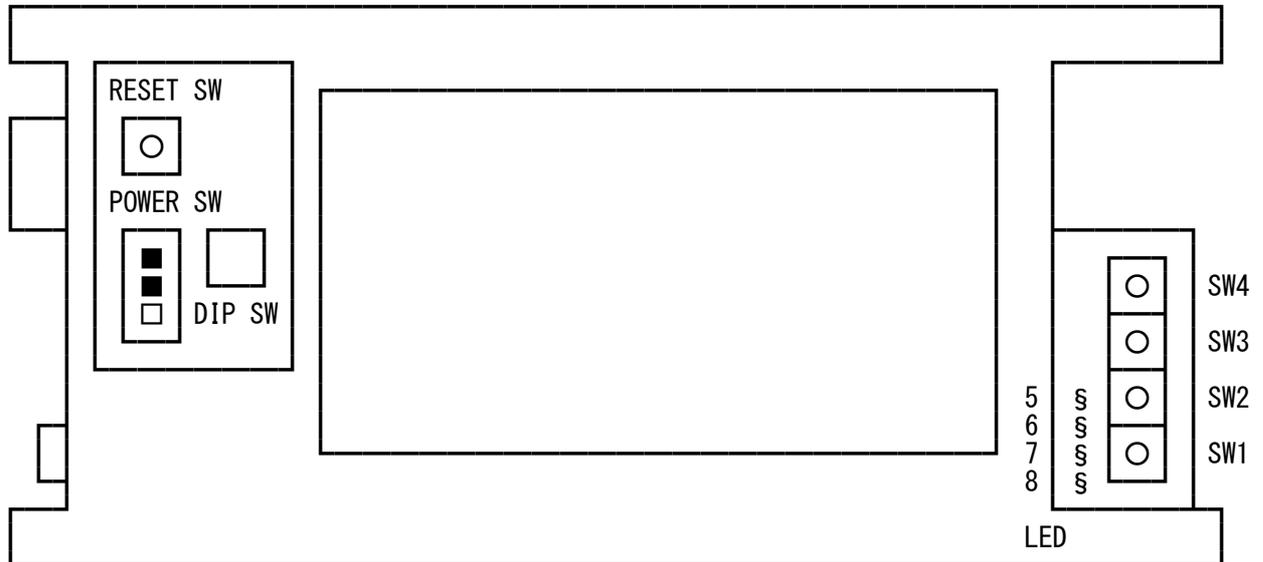
T-Kernel 2.0 リファレンスキット上で動作する T-Kernel オペレーティングシステムとアプリケーションおよびツールなどの実行プログラムが含まれています。PMC T-Shell 環境での追加システムが、CD-ROM に入っています。

**ホスト側**

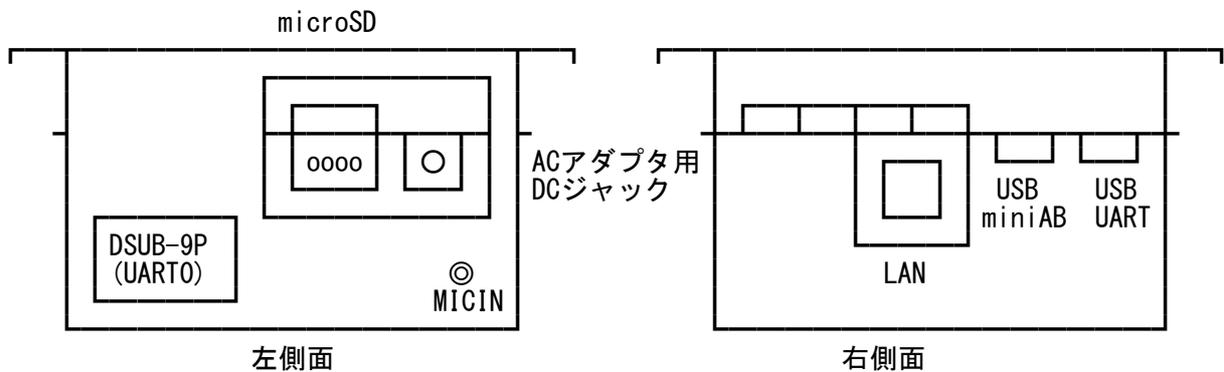
本製品で使用する GNU 開発環境と、デバイスドライバやサンプルアプリケーションなどのソースが含まれています。開発用ホストマシンとして Linux または、Windows 上の Cygwin 環境を使用します。

## 1.2 ハードウェア取り扱い方法

T-Kernel 2.0 リファレンスキットには、下図に示す各種のスイッチ/コネクタが付いています。ここでは、使用する上で必要となる事項を簡単に説明します。詳細は「T-Engine Reference Board Hardware Manual」を参照してください。



正面



左側面

右側面

## ○ DIP SW

4つの DIPSW の設定内容は、以下の通りです。設定は必ず電源 OFF の状態で行ってください。

DIPSW は、出荷時はすべて OFF になっています。

DIPSW-1、2、3 は、OFF 固定です。変更しないでください。

DIPSW-4 は、OFF で自動ブート (OS 起動)、ON でブートせずに T-Monitor が起動します。

	OFF	ON
DIPSW-1 BOOTSEL0	OFF	-
DIPSW-2 BOOTSEL1	OFF	-
DIPSW-3 BOOTSEL2	OFF	-
DIPSW-4 起動選択	自動ブート	T-Monitor 起動

- RESET SW  
システムをリセットして再起動します。リセットが必要になったときは、このボタンを押してください。
- POWER SW  
ACアダプタ用DCジャック、または電源供給コネクタから電源を供給し、POWER SW を入れるとシステムが起動します。
- LED5, 6, 7, 8  
LED が 4個あり、T-Monitor のプログラムサポート機能を使用して、プログラムで点灯させることができます。
- SW1, 2, 3, 4  
KB/PD ドライバを通して、キーとして使用することが出来ます。SW1 = →、SW2 = '3'、SW3 = Enter、SW4 = '1' になります。  
SW1 を押しながら起動すると、DIPSW-4 が OFF の状態(自動ブート)でも T-Monitor が起動します。
- microSD スロット  
システムディスク用の microSD カードスロットです。
- USB mini AB コネクタ  
USB ホストのコネクタです。USB 2.0 に対応しています。
- USB UART コネクタ  
内部で USB-UART 変換チップに接続され、UART2 として通信が可能です。
- DSUB-9P(UART0) コネクタ  
DSUB-9P は T-Engine リファレンスボードの CN3 に接続されており、デバッグコンソール接続用のコネクタです。制御信号線はありません。
- AC アダプタ用DCジャック  
キット付属の AC アダプタ (5V) を接続します。なお、機器に組み込む場合等、コネクタが容易に抜けては困る場合には、ACアダプタ用DCジャックのかわりに電源供給コネクタ (XHコネクタ) を使用できます。詳細はハードウェアマニュアルを参照してください。ACアダプタを接続する場合、電源供給コネクタに電源を接続しないでください。
- LAN コネクタ  
10BASE-T、または 100BASE-TX の LAN が接続できます。

### 1.3 T-Kernel 2.0 リファレンスキットの起動方法

#### (1) デバッグコンソールの準備

DSUB-9P コネクタに RS-232C クロスケーブルを接続して、他方を開発用ホストマシンに接続し、開発用ホストマシン上で通信ソフトウェアを起動してください。

デバッグコンソールの通信仕様は以下の通りです。

通信速度	115,200 bps
データ長	8 ビット
ストップビット	1 ビット
パリティ	なし
フロー制御	XON/XOFF (ハードフロー制御(RS/CS)は利用不可)
文字コード	ASCII / EUC コード
受信行末	CR (0x0d)
送信行末	CRLF (0x0d, 0x0a)

#### (2) AC アダプタ用DCジャックに付属の AC アダプタを接続してください。

AC アダプタを接続し、POWER SW をRESET SW の方向にスライドすると電源 ON となります。システムが起動して、開発用ホストマシン上のターミナルソフトウェアに起動メッセージが表示されます。

DIP SW-4 が OFF のときは、ROM ディスクからブートして GLI のコマンド待ちの状態となります。ON のときは、T-Monitor が立ち上がり、T-Monitor のコマンド待ちの状態となります。T-Monitor のコマンド待ちから BootDisk (BD) コマンドを入力することにより、ディスクからブートできます。

### 1.4 USB での接続

何らかの事情でシリアルポートをデバッグコンソールとして使用できない場合、USB UART コネクタの UART2 をデバッグコンソールにすることが出来ます。

#### 1.4.1 はじめて Windows ホストに USB 接続する場合

はじめて Windows ホストに USB 接続する場合、デバイスドライバのインストール作業が必要になります。そのため、管理者として Windows にログオンしてください。

デバイスドライバは CD-ROM 内の common/soft/CP210x\_VCP\_Windows.zip です。

※)Silicon Laboratories 社のホームページから、デバイスドライバをダウンロードすることも可能です。

[http://www.silabs.com/Support%20Documents/Software/CP210x\\_VCP\\_Windows.zip](http://www.silabs.com/Support%20Documents/Software/CP210x_VCP_Windows.zip)

上記のアドレスを Webブラウザでアクセスすると、ファイル保存のダイアログが表示されますので、保存します。後の操作は CD-ROM の場合と同様です。

デバイスドライバのファイルを右ボタンクリックして「すべて展開」を実行します。展開したディレクトリの中に CP210xVCPInstaller\_x86.exe と CP210xVCPInstaller\_x64.exe があります。32bit システムの場合は前者を、64bit システムの場合は後者をダブルクリックしてください。「発行元を確認できませんでした。このソフトウェアを実行しますか?」と確認を求めてきますので、「実行」を押してください。インストーラーが起動しますので、表示に従ってインストールを行ってください。インストールが完了した後、USBケーブルを T-Engine リファレンスボードの USB UART コネクタに接続し、もう一方を開発用ホストマシンに接続してください。自動的にデバイスドライバがインストールされます。

デバイスドライバ登録後、割り当てられた COM ポートを確認します。

デバイスマネージャを開きます。

Windows XP の場合

- 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-システム-ハードウェア-デバイスマネージャ で開きます。

Windows 8.1/8/7/Vista の場合

- 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 で開きます。

ポート(COM と LPT)に「Silicon Labs CP210x USB to UART Bridge (COMx)」が表示されることを確認してください。COMx が割り当てられた COM ポートになります。

## 1.4.2 接続方法(シリアル/USB)の変更

### (1)一時的に UART2 をデバッグコンソールにしたい場合

SW2 を押しながら 電源を入れる、または、SW2 ボタンを押しながら RESET SW を押すと、次に再起動するまで UART2 をデバッグコンソールにすることが出来ます。

### (2)恒久的に UART2 をデバッグコンソールにしたい場合

どの UART をデバッグコンソールにするかは ROM 情報に基づいて決定されます。ROM 情報を書き換えることでデバッグコンソールとして使用する UART を変更することが出来ます。

ROM 情報のイメージをCD-ROM内に用意してあります。

jp/soft/rominfo.mot	UART0 (DSUB-9Pコネクタ)をデバッグコンソールとして使用する ROM 情報(標準設定)
jp/soft/rominfo-usb.mot	UART2 (USB-UART)をデバッグコンソールとして使用する ROM 情報

これらのファイルを使用して、フラッシュメモリを書き換えてください。具体的な方法は後述の「フラッシュメモリの再書き込み」を参照してください。

- † 恒久的に UART2 をデバッグコンソールにした状態で SW2 を押しながら 電源を入れる、または、SW2 ボタンを押しながら RESET SW を押すと、次に再起動するまで DSUB-9Pコネクタ(UART0) をデバッグコンソールにすることが出来ます。

### 1.4.3 USB デバッグポートを使用する際の注意事項

#### (1) 接続方法に関する注意事項

開発用ホストマシンと T-Kernel 2.0 リファレンスキットは直接接続する事を推奨します。USBハブを介して接続すると不具合が発生する場合があります。

#### (2) T-Kernel 2.0 リファレンスキット起動・終了時における注意事項

製品付属の通信ソフト gterm 以外の通信ソフトをご利用の場合、通信中に T-Kernel 2.0 リファレンスキットの電源を落とすと、その後電源を再度投入しても通信が出来ない場合があります。そのような場合は、一度通信ソフトを終了させて再度、通信ソフトを起動してください。

## 1.5 起動ディスクの作成

### 1.5.1 基本機能

フラッシュメモリ上の ROM ディスクシステムは、基本システムのみを含んだ状態になっています。また、読み出し専用ですので、開発したソフトウェアを直接格納することが出来ません。

開発作業を始める前に、まず、書き込み可能なメディアに起動用ディスクを作成してください。書き込み可能なメディアとして、以下のデバイスが利用できます。

#### (1) microSD

デバイス名: pcb

#### (2) eMMC

デバイス名: pcc

#### (3) USBストレージデバイス (USB Mass Storage Class 仕様準拠のもののみ)

デバイス名: uda

実際にディスクを使用するためには、CLI 上で区画設定、およびフォーマットを行う必要があります。以下に microSD の場合を例に起動用ディスクを作成する手順を示します。

T-Kernel 2.0 リファレンスキットの microSD カードスロットに microSD カードを挿入し、ACアダプタから電源を投入します。

T-Kernel 2.0 リファレンスキット が起動したら以下の手順で区画を作成します。

(重要) 以下の操作を行うとディスクの既存の内容は破壊されますのでご注意ください。  
Boot 区画の設定を忘れないでください。

```
[/SYS]% hdpart pcb
pcb  [C:485 H:64 S:63 B:1961984 (958 MB)]
No System  Boot  StartCHS      EndCHS          SecNo   SecCnt   Size
1  0b DOS    80      0: 1: 1      486: 38:38      63     1961921  957 MB
2  00 -----  00      0: 0: 0      0: 0: 0         0         0      0 KB
3  00 -----  00      0: 0: 0      0: 0: 0         0         0      0 KB
4  00 -----  00      0: 0: 0      0: 0: 0         0         0      0 KB
** Create/Delete/Boot/Edit/Quit ? d
```

```

Delete PartNo (1-4, All) ? 1
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/Update/Quit ? c
Create PartNo (1-4) ? 1
Size [GB/MB/KB, All] (<958MB) ? a
No System  Boot  StartCHS  EndCHS  SecNo  SecCnt  Size
1 13 BTRON  00  0: 1: 1  486: 38:38  63  1961921  957 MB
2 00 ----- 00  0: 0: 0  0: 0: 0  0  0  0 KB
3 00 ----- 00  0: 0: 0  0: 0: 0  0  0  0 KB
4 00 ----- 00  0: 0: 0  0: 0: 0  0  0  0 KB
** Create/Delete/Boot/Edit/Update/Quit ? b
Boot PartNo (1-4, Clear) ? 1
No System  Boot  StartCHS  EndCHS  SecNo  SecCnt  Size
1 13 BTRON  80  0: 1: 1  486: 38:38  63  1961921  957 MB
2 00 ----- 00  0: 0: 0  0: 0: 0  0  0  0 KB
3 00 ----- 00  0: 0: 0  0: 0: 0  0  0  0 KB
4 00 ----- 00  0: 0: 0  0: 0: 0  0  0  0 KB
** Create/Delete/Boot/Edit/Update/Quit ? u
** pcb: Updated Master Boot Block
[/SYS]%

```

† 他の用途に使用中の区画にインストールする場合は、最初にd>Delete) コマンドで対象の区画を削除した後で、c>Create) の操作を行ってください。

続いて format コマンドを使用してフォーマットを行います。引数に -b を指定し、ブートブロックを書き込みます。

フォーマットする区画をデバイス名で指定します。先頭区画が pcb0、2番目の区画が pcb1 のようになりますので、hdpart で指定したデバイス名の後ろに番号を付けて指定してください。デバイス名に付ける区画番号(サブユニット番号)は 0 から始まります。hdpart での区画番号とは異なりますので注意してください。

(重要) 区画番号(サブユニット番号)を付けずにフォーマットすると、ディスク全体のデータが失われます。必ず区画番号を付けたデバイス名を指定してください。

```

[/SYS]% format -b pcb0 SYSTEM
Format pcb0 [STD] SYSTEM
Logical Formatting...
Writing BootCode...
Disk Format Success.
[/SYS]%

```

† 2 GB 以上の区画の場合、format コマンドで -x 指定を行ってください。

```
[/SYS]% format -b -x pcb0 SYSTEM
```

・ファイルシステム名(上記例では SYSTEM)は、任意に指定してください。

なお、同じファイルシステム名を同時に複数使用することはできません。複数のディスクまたは区画を使用する場合は、それぞれ異なるファイルシステム名とする必要があります。

フォーマットが済んだメディアに対し、ROM ディスク上のプログラムをコピーします。以下の手順でコピーして下さい。

```
[/SYS]% att pcb0 /A
[/SYS]% rcp -r /SYS /A/=
```

†コピーには少し時間がかかります。

作成した起動ディスクを切断し、再起動します。

```
[/SYS]% det pcb0
[/SYS]% exit
[IMS]% exit -1
```

```
<< START CLI >>
20xx/xx/xx(x) xx:xx:xx
[/SYS]%
```

†USB ストレージから起動したい場合、追加の操作が必要です。

USB ストレージから起動する場合でも、起動対象となるデバイスの検索順序(実装仕様書参照)にそって検索されたディスクから起動します。その後、その検索されたディスク内に記述された設定に従って、起動ディスクが USB ストレージに切り替わります。その際に、以下のファイルは 検索されたディスク内のものが使用され、USB ストレージ内の同名のファイルは使用されません。

PBOOT, SBOOT, KERNEL.SYS, SYSCONF, DEVCONF, STARTUP.CMD, chgsys

デフォルトでは USB ストレージからは起動しない設定となっておりますので、USB ストレージから起動したい場合は検索されるディスク内の STARTUP.CMD を修正してください。検索されるディスクが ROM ディスクの場合、ROM ディスクの書き換えが必要になります。ROM ディスクの書き換えは「ROM ディスクの更新」を参照してください。

作成した起動ディスク内の SYSCONF を修正することにより、リソースの設定などを変更することができます。また、STARTUP.CMD や STARTUP.CLI にプログラムを追加することで、起動時に自動的に実行させることができます。

### 1.5.2 PMC T-Shell の追加

GUI ミドルウェアを使用する場合は、引き続き PMC T-Shell をインストールします。以下に起動ディスクに PMC T-Shell をインストールする手順を説明します。

- (1) T-Kernel 2.0 リファレンスキット を起動ディスク(書き込み可能なメディア上に作成したもの)から起動します。
- (2) 通信ソフトの XMODEM ファイル転送機能を使用してインストールファイルを転送します。

```
[/SYS]% recv -d -c install_tshell.bz
```

† recv コマンドを実行するとターゲットのCLI(コマンドラインインタプリタ)が転送待ちになりますので、通信ソフトの XMODEM 送信機能を使って、CD-ROM の jp/soft/install\_tshell.bz を転送します。

† T-Kernel 2.0 リファレンスキット の開発環境に付属している通信ソフト「gterm」を使用している場合、カレントディレクトリに install\_tshell.bz があれば自動的に転送を開始します。カレントディレクトリにファイルが無い場合はフルパスで転送するファイルを指定してください。

† 開発用ホストマシン側でftpサーバが動いており、T-Kernel 2.0 リファレンスキットと LAN 経由での接続が可能であれば、recvコマンドでシリアル経由の転送をする代わりに、fget コマンドによる LAN 経由の転送をすることも可能です。

† USB A-typeコネクタを、USB mini-A コネクタに変換するケーブルがあれば、T-Kernel 2.0 リファレンスキット に FAT フォーマットの USB メモリを接続することが可能です。まず、Windows 上で USB メモリに CD-ROM の jp/soft/install\_tshell.bz をコピーします。T-Kernel 2.0 リファレンスキット の USB mini AB コネクタに変換ケーブルを介して USB メモリを接続し、以下のコマンドでファイルをコピーしてください。

```
[/SYS]% att -m uda0 uda0
[/SYS]% ux/cp /uda0/install_tshell.bz .
[/SYS]% det -u uda0
```

### (3) インストールファイルを展開します。

```
[/SYS]% expf -r install_tshell.bz
```

† しばらく時間がかかります。

### (4) 起動ディスクに PMC T-Shell のインストールを行います。

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

† しばらく時間がかかります。

† インストールを行うと、以下のシステムファイルが更新され、元のファイルはファイル名に .bak がついた形で保存されます。これらのファイル内容を修正していた場合は、再度修正を行う必要がありますので注意してください。

SYSCONF	システムコンフィグレーション情報
DEVCONF	デバイスコンフィグレーション情報
STARTUP.CMD	IMS 起動スクリプト
STARTUP.CLI	CLI 起動スクリプト

### (5) インストールが完了したら、以下のコマンドを実行して不要なインストールファイルを削除し、T-Kernel 2.0 リファレンスキット を再起動してください。

```
[/SYS]% rm -r install_tshell.bz install_tshell
```

```
[/SYS]% exit  
[IMS]% exit -1
```

### 1.5.3 PMC T-Shell インストール後の動作確認

PMC T-Shell をインストールした起動ディスクから、T-Kernel 2.0 リファレンスキット を起動すると、デバッグコンソールに一連のメッセージを表示して、画面にウィンドウが表示されます。

T-Kernel 2.0 リファレンスキット 上で表示される画面(ウィンドウ)の操作は、超漢字V(別売)と基本的に同じですが、タッチパネル用に、画面下のシステムメッセージパネル部分をプレスするとメニューが表示されるように変更されています。その他の操作方法につきましては 超漢字V(別売)の説明書を参照してください。

†ただし、本製品に含まれるプログラムは、超漢字V(別売)に存在するプログラムの内の一部です。そのため、超漢字V(別売)に比べて、メニュー項目が少なかったり、機能が実行できない部分があります。

メニューの「小物」「タッチパネル調整」を選択することにより、タッチパネルの座標位置の調整を行うことができます。画面上に表示される指示にしたがって、操作してください。調整した結果は、すぐに反映されず、再起動しないと反映されませんので注意してください。

†作成した起動ディスクでは、デバッグ用のためポインタ表示を行うようになっていますが、「タッチパネル調整」により、ポインタ表示を行わないようにできます。

インストールした起動ディスクには、マイクロスクリプト以外にもいくつかの 超漢字V(別売)から移植したアプリケーションが含まれています。

## 1.6 フラッシュメモリの再書き込み

出荷時には、フラッシュメモリにシステムが書き込んでありますが、再書き込みが必要になった場合は、T-Monitor のコマンドを使用して、以下の手順でフラッシュメモリに再書き込みしてください。

- (1) DIP SW-4 を ON にして、システムを起動して T-Monitor を立ち上げてください。
- (2) FlashLoad (FLL0) コマンドを入力してください。

```
TM> FlashLoad
```

- (3) 通信ソフトのファイル送信機能を使用して、S フォーマット形式のデータファイルを無手順テキスト形式で転送してください。

システム全体を初期状態に戻したい場合は、CD-ROM 内の以下のファイルを無手順テキスト形式で転送してください。

```
jp/soft/romimage.mot (S フォーマット形式)
```

†FlashLoad のコマンドを入力後、下記の行が表示された後に通信ソフトによるファイル転送を開始してください。この行が表示される前に転送を開始するとデータを取りこぼして

しまいます。

> Load S-Format Data of Flash ROM

†開発キットに付属している通信ソフト gterm を使用している場合は、(2)(3)の手順は、以下の 1 つのコマンドで行うことができます。

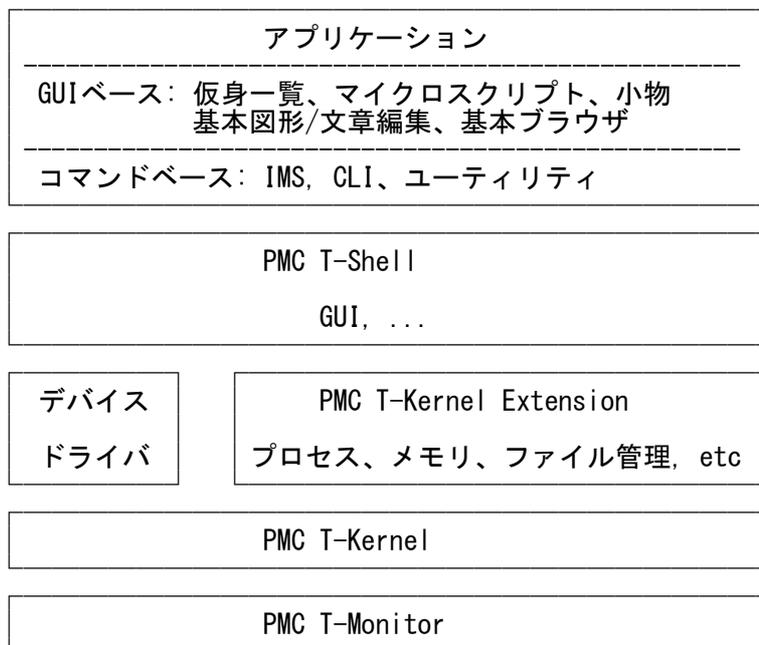
```
TM> .flload jp/soft/romimage.mot
```

- (4) 転送が終了すると、自動的にフラッシュメモリへの書き込みが行われ、必要に応じてシステムが再起動します。

## 2. ソフトウェア構成

### 2.1 全体構成

本製品の全体のソフトウェア構成を以下に示します。



#### ○ PMC T-Monitor

T-Monitor仕様書に準拠したモニタです。詳細は「T-Monitor 仕様書」および「実装仕様書」を参照してください。

#### ○ PMC T-Kernel

T-Kernel仕様書に準拠したリアルタイムOSです。T-Kernel/OS, T-Kernel/SM, T-Kernel/DSを含んでいます。詳細は「T-Kernel 2.0仕様書」および「実装仕様書」を参照してください。

#### ○ PMC T-Kernel Extension

本製品用に、T-Kernel/OS のサブシステム機能を利用して構築した OS の拡張部分で、BTRON3 仕様に準拠した以下の機能を含んでいます。詳細は「PMC T-Kernel Extension 説明書」を参照してください。

プロセス/タスク管理、メッセージ管理、プロセス/タスク間同期通信管理、グローバル名管理、メモリ管理、ファイル管理、イベント管理、デバイス管理、時計管理、システム管理、UNIX(ファイル)エミュレータ

この拡張により、仮想記憶 OS を実現しており、開発環境としてファイルやプロセスを利用できるようになっています。

## ○ PMC T-Shell

本製品用に、T-Kernel/OS のサブシステム機能を利用して構築したミドルウェア群であり、BTRON3 仕様に準拠した以下の機能を含んでいます。GUI を使用したアプリケーションの開発のための多くの機能を提供しています。

ディスプレイプリミティブ、フォントマネージャ  
フォントデータ (ドット、TrueType)  
ウィンドウマネージャ、メニューマネージャ、パーツマネージャ、パネルマネージャ、トレーマネージャ、データマネージャ  
テキスト入力プリミティブ  
実身/仮身マネージャ

## ○ デバイスドライバ

本製品用に、T-Kernel/SM デバイス管理機能に基づいた以下のデバイスドライバが含まれています。デバイスドライバの詳細は「デバイスドライバ共通説明書」、「ネットワークデバイスドライバ説明書」、「USBマネージャ取扱説明書」および「実装仕様書」を参照してください。

USB マネージャ (バスドライバ)  
時計 (RTC)、コンソール (シリアル)、システムディスク (microSD、eMMC、USBメモリ)、KB/PD (タッチパネル、USB KB/MOUSE)、スクリーン、LAN

## ○ アプリケーション

下記のコマンドベースの開発用ツールが含まれています。

IMS (Initial Monitor System)  
詳細は「開発ツール説明書」の「IMS」の章を参照してください。  
CLI (Command Line Interpreter)  
詳細は「開発ツール説明書」の「CLI」の章を参照してください。  
フォーマッタ、その他各種ツール、テストプログラムなど  
詳細は「開発ツール説明書」の「ユーティリティ」の章を参照してください。

下記の GUI アプリケーションが含まれています。

仮身一覧、マイクロスクリプト、基本図形/文章編集、基本ブラウザ、各種小物  
詳細は超漢字V (別売) に含まれる電子マニュアルを参照してください。

## 2.2 ファイル構成

PMC T-Shell までインストールした起動ディスクには、以下のファイルが格納されています。

SBOOT	2次ブートプログラム
KERNEL.SYS	OS 核 ( T-Kernel, Extension, ドライバ一部)
SYSCONF	システムコンフィグレーションファイル (EUC テキスト)
DEVCONF	デバイスコンフィグレーションファイル (EUC テキスト)
STARTUP.CMD	システム起動コマンド (EUC テキスト)
STARTUP.CLI	CLI 起動コマンド (EUC テキスト)
chgsys	システムディスク切り替え
screen	画面(スクリーン)ドライバ
kbpd	KB/PD ドライバ
lowkbpd	KB/PD 実 I/O ドライバ
rsdrv	RS ドライバ
netdrv	LAN ドライバ
unixemu	UNIX(ファイル)エミュレータ
tcpipmgr	TCP/IP マネージャ
NETCONF	ネットワークコンフィグレーション情報
cli	CLI
.xcli	CLI コマンドファイル (EUC テキスト)
starting	初期画面イメージ
lib/	共有ライブラリ
libapp. so. 2	
libbt. so. 2	
libdl. so. 2	
libg. so. 2	
libimg. so. 2	
libstlport. so. 2	
libux. so. 2	
libwin. so. 2	
libtf. so. 2	
bin/	CLI ユーティリティ
dd	ディスクダンプツール
ed	簡易行エディタ
cmp	ファイル比較
expf	解凍ツール
hdpart	簡易ディスク区画作成
format	ディスクフォーマット
devlist	デバイス一覧
usbinfo	USB デバイス情報表示ツール
mscnv	簡易ファイル変換ツール
debugmode	DEBUGMODE 変更ツール
devconf	DEVCONF 変更ツール
sysconf	SYSCONF 変更ツール
netconf	NETCONF 作成ツール
netdump	LAN パケット情報の表示(テスト用)
netinf	LAN ドライバ情報の表示(テスト用)

netstart	ネットワーク起動ツール
ping	簡易 ping ツール
telnet	簡易 telnet ツール
fget	ftp によるファイルのダウンロード
fput	ftp によるファイルのアップロード
ppp	PPP 接続ツール
ux/	UNIX(ファイル)エミュレータコマンドツール
pwd	カレントディレクトリのパス名表示
ls	ディレクトリ内容のリスト表示
cp	ファイル(ディレクトリ)のコピー
rm	ファイル(ディレクトリ)の削除
mv	ファイル(ディレクトリ)の移動、ファイル名変更
mkdir	ディレクトリの作成
rmdir	ディレクトリの削除
chmod	ファイルアクセス権の変更
fcnv	ファイル変換
chver	バージョン変更ツール
chattr	ファイル属性変更ツール
vup	ファイルバージョンアップツール
pacf	アーカイブ作成ツール
fbox	\$\$FONT.BOX ツール
font	フォントマネージャ
font_nottf	フォントマネージャ (TrueType 未対応版)
fpsvr	フォントサーバー
dp	ディスプレイプリミティブ
tip	テキスト入力プリミティブ
hmi	HMI マネージャ
omgr	実身/仮身マネージャ
sysdmn	システムデーモンプロセス
logon	ログオンプロセス
\$\$SYSDBOX	システムデータボックス
@USER_ID_CARD	ユーザ情報
\$\$BGSCREEN.BOX	背景画面
\$\$CUSTOMIZE.BOX	画面カスタマイズデータ
\$\$RELATION.BOX	続柄情報
\$\$PROGRAM.BOX/	プログラム格納場所
DLED	仮身一覧
CHGENV	ユーザ環境設定
SYSENV	システム環境設定
MSCRIPT	マイクロスクリプト
CALC	電卓
CLK	時計
TXED	基本文章編集
FGED	基本図形編集
bbb	基本ブラウザ
NETENV	ネットワーク設定
FINDER	仮身参照
FSEARCH	仮身検索

	NET	仮身ネットワーク
	FORMATTER	ディスク初期化
	MANMOS	原紙箱
	GADGET	小物箱
	DEVICE	ディスク集め
	TPADJ	タッチパネル調整
	CONSOLE	コンソール
FONT/		基本フォント
	基本-明朝固定S	
\$\$FONT.BOX/		追加フォント
	基本-明朝可変	
USR/		初期ウィンドウ
	原紙箱	
	小物箱	
	ブラウザ用紙	
	キャビネット	
	原稿用紙	
	画用紙	
	スクリプト用紙	
	小物入れ	
	サンプル集	

### 2.3 システムの起動手順

システムの起動は以下の手順で行われます。

- (1) T-Monitor が起動対象ディスクのブートブロックを読み込んで、実行します。ブートブロックには、1 次ブートプログラム (PBOOT) が格納されています。
- (2) PBOOT は、ファイルシステム内の 2 次ブートプログラム (SBOOT) を捜して、メモリに読み込んで実行します。
- (3) SBOOT は、ファイルシステム内の以下のファイルをそれぞれメモリに読み込んで、KERNEL.SYS を実行します。

KERNEL.SYS	OS 核 ( T-Kernel, Extension, ドライバー部)
SYSCONF	システムコンフィグレーションファイル
DEVCONF	デバイスコンフィグレーションファイル

- (4) KERNEL.SYS は、SYSCONF で指定された各種のパラメータにしたがって、初期化を行い、PMC T-Kernel および PMC T-Kernel Extension を動作させます。さらに含まれている以下のドライバも立ち上げます。これらのドライバは以降のシステムの立ち上げに必須のため、KERNEL.SYS に含まれています。

USB マネージャ, 時計 (RTC), コンソール (シリアル), システムディスク

- (5) OS としてのすべての初期化が終了すると、初期タスクとして IMS が立ち上がり、STARTUP.COMD ファイルを読み込んで、その内容にしたがった処理を行います。

- (6) STARTUP.CMD では、他のドライバ、サブシステム、サーブプロセスなどの基本部分を立ち上げたのち、最後に STARTUP.CLI をパラメータとして、CLI を実行します。
- (7) CLI は、STARTUP.CLI ファイルの内容を読み込んで、その内容にしたがって処理を行います。
- (8) STARTUP.CLI では、/SYS/USR を初期ウィンドウとして、仮身一覧を起動します。

### 3. システムコンフィグレーション

#### 3.1 GUI 機能

GUI 機能は、大きく 3 つのレベルに分類できます。

インストール直後は、(3)BTRONシステムレベルになっていますが、利用レベルに応じて不要なモジュールを削除して、STARTUP.CMD 内の対応する起動行を削除またはコメントアウトしてください。

† STARTUP.CMD は、T-Kernel 2.0 リファレンスキット上で簡易エディタの ed ツールを使用して直接編集するか、または、開発環境の \$BD/kernel/config/em1d512.te にインストールされる STARTUP.CMD ファイルを開発用ホストマシン上で編集してダウンロードしてください。

† \$BD は開発環境のベースディレクトリを表します。通常、コマンドライン版の開発環境の場合は /usr/local/te になります。Eclipse 版の場合 C:\eclipse\plugins\com.t\_engine4u.tl.em1d512\_te.x.y.z\_x.y.z\te になります。

##### (1) 基本描画レベル

画面描画のための基本機能レベルで、ディスプレイプリミティブにより実現されます。

/SYS/dp	ディスプレイプリミティブ
---------	--------------

文字描画を行うためには、フォントマネージャとフォントデータが必要になります。文字描画を行わない場合は、削除可能です。

/SYS/font	フォントマネージャ
/SYS/fpsvr	フォントサーバー
/SYS/FONT/*	基本フォントデータ (必須)
/SYS/\$\$FONT.BOX/*	追加フォントデータ (オプション)

##### (2) ウィンドウレベル

ウィンドウシステムのレベルで、以下の機能を 1 つにまとめた HMI マネージャにより実現されます。このレベルの実現のためには、(1) 基本描画レベルは必須となります。

ウィンドウマネージャ	
メニューマネージャ	
パーツマネージャ	
パネルマネージャ	
トレーマネージャ	
データマネージャ	
/SYS/hmi	HMI マネージャ
/SYS/\$\$SYSDBOX	システムデータボックス
/SYS/tip	テキスト入力プリミティブ

##### (3) BTRONシステムレベル

実身/仮身を使用した、BTRONシステムレベルは、以下のモジュール/データにより実現されます。このレベルの実現のためには、(1)基本描画レベル、および(2)ウィンドウレベルは必須となります。

/SYS/omgr	実身/仮身マネージャ
/SYS/logon	ログオンプロセス
/SYS/\$\$RELATION. BOX	続柄情報
/SYS/\$\$PROGRAM. BOX/*	プログラム格納場所

### 3.2 ユーザ環境

「ユーザ環境設定」小物で設定した、ユーザの操作環境に関する各種の設定は、以下のファイルに保存されています。

/SYS/@USER_ID_CARD	ユーザID カード
--------------------	-----------

### 3.3 画面カスタマイズ

画面のカスタマイズに関して、以下のデータファイルが利用できます。

\$\$BGSCREEN. BOX	背景画面(壁紙)
\$\$CUSTOMIZE. BOX	画面のカスタマイズ(配色/書体)情報

インストール直後は、サンプルの背景画面が入っています。配色/書体は、標準のみの状態です。

「システム環境設定」小物の「画面-背景/配色/書体」により、背景画面や、画面のカスタマイズ(配色/書体)を行うことができます。

### 3.4 フォント

フォントの機能は、以下のモジュールにより実現されます。

/SYS/font	フォントマネージャ
/SYS/fpsvr	フォントサーバー
/SYS/FONT/*	基本フォントデータ(必須)
/SYS/\$\$FONT. BOX/*	追加フォントデータ(オプション)

フォントを利用しない場合は、上記モジュールをすべて削除し、STARTUP.CMD 内の font の起動行を削除またはコメントアウトしてください。

また、ドットフォントのみを使用し、TrueType フォントを使用しない場合は、font\_nottf (font よりサイズが小さい) を /SYS/font の代わりに利用することができます。この場合でも fpsvr は必要です。

インストール直後は、以下のフォントデータが登録されています。

/SYS/FONT/基本-明朝固定S
--------------------

12, 16ドットの漢字フォント

8, 12, 16ドットの英数全角/英数半角/英数比例ピッチフォント

/SYS/\$\$FONT.BOX/基本-明朝可変

明朝体 TrueType フォント

CD-ROM 内の common¥font ディレクトリには、以下のフォントが含まれています。

† フォントの内容は、超漢字V(別売)に含まれる「システム環境設定」小物で「書体」の詳細情報を参照してください。T-Kernel 2.0 リファレンスキット上で「システム環境設定」小物では、「書体」の詳細情報は表示されません。

基本-明朝固定S  
基本-明朝固定  
基本-明朝16  
基本英数-明朝固定  
補助-明朝固定  
補助非漢字-明朝固定  
中国-明朝固定  
韓国-明朝固定  
点字-固定  
基本-明朝可変  
基本-SS明朝可変  
基本-SS細明朝可変  
基本-SSゴシック可変  
基本-SS教科書可変  
基本-SS丸ゴシック可変  
基本英数-SSAvalon可変  
基本英数-SSCourier可変  
基本英数-SSEuromode可変  
基本英数-SSHelvetica可変  
基本英数-SSHelveticaN可変  
基本英数-SSTimes可変  
補助-明朝可変  
補助-SS明朝可変  
補助-SSゴシック可変  
日本第3第4-明朝可変  
GT-GT書体可変  
GT-GT書体追加可変  
大漢和-SS細明朝可変  
中国-SS明朝可変  
中国-SSゴシック可変  
中国-SS傍宋可変  
中国-SS楷書可変  
韓国-SS明朝可変  
韓国-SS明朝可変太字  
韓国-SSゴシック可変  
韓国-SSゴシック可変太字

韓国-SS宮書可変  
 韓国-SS丸ゴシック可変  
 台湾-SS明朝可変  
 台湾-SSゴシック可変  
 各国-SS明朝可変  
 各国-SSゴシック可変  
 シンボル-iモード絵文字可変  
 シンボル-序数記号可変  
 シンボル-陰陽五行可変  
 シンボル-ホツマ文字可変  
 点字-印刷用可変  
 点字-表示用可変

フォントの登録方法/削除方法に関しては、CD-ROM の common¥font ディレクトリ内の readme.txt を参照してください。なお、/SYS/FONT に入っている基本フォントは、変更しないでください。

### 3.5 ネットワーク

ネットワークの機能は、以下のモジュールにより実現されます。

/SYS/netdrv	LAN ドライバ
/SYS/tcpipmgr	TCP/IP マネージャ
/SYS/NETCONF	ネットワークコンフィグレーションファイル

ネットワークを全く利用しない場合は、上記モジュールをすべて削除し、STARTUP.COMD 内の netdrv、および tcpipmgr の起動行を削除またはコメントアウトしてください。

ネットワークコンフィグレーションファイル (NETCONF) には、ホスト、DNS サーバ、ドメイン名、ゲートウェイなど設定します。「ネットワーク設定」小物で設定しますが、CLI 上で netconf ツールを利用して設定することもできます。

```

[/SYS]% netconf          # NETCONF の表示
hostname =
host ip   = 0.0.0.0
dns1name =
dns1 ip   = 0.0.0.0
dns2name =
dns2 ip   = 0.0.0.0
domain   =
gateway ip = 0.0.0.0
subnetmask = 255.255.255.0

[/SYS]% netconf c        # NETCONF の作成/変更
hostname = ? myhost
host ip   = 0.0.0.0 ? 198.162.0.2
dns1name = ? dns1
dns1 ip   = 0.0.0.0 ? 198.162.0.254
dns2name = ?
  
```

```

dns2 ip    = 0.0.0.0 ?
domain     = ?          tshell-test
gateway ip = 0.0.0.0 ?
subnetmask = 255.255.255.0 ?
wlan       = none (n/a/i)? n # n:noe, a:adhoc, i:infra

```

† wlan は、無線LAN の設定ですが、T-Kernel 2.0 リファレンスキットでは、無線LAN は利用できません。

† netconf ツールは、NETCONF ファイルが存在しないときは自動的に生成しますので、設定がうまくいかないときは、一度 NETCONF ファイルを削除してから設定し直してください。

```

[/SYS]% rm NETCONF
[/SYS]% netconf c

```

インストール直後の NETCONF は、空（ホスト IP アドレス = 0.0.0.0）の設定となっていますが、接続したネットワーク上に DHCP サーバが存在する場合は、DHCP サーバから IP アドレスを取得するため、そのまま動作させることが可能です。

ネットワーク機能に関しては、以下のユーティリティが用意されています。それぞれ、パラメータなしで実行すると、簡単なヘルプメッセージが表示されます。

ping	簡易 ping ツール
fget	ftp によるファイルのダウンロード
fput	ftp によるファイルのアップロード
telnet	簡易 telnet ツール
ppp	PPP 接続ツール
netstart	ネットワーク起動ツール
netinf	LAN ドライバ情報の表示(テスト用)
netdump	LAN パケット情報の表示(テスト用) netdump ? で、ヘルプメッセージが表示されます。

ネットワーク機能を利用したプログラミングに関しては、「PMC T-Shell 説明書」の「TCP/IP マネージャ」の章を参照してください。BSD ソケット I/F に準拠した API が用意されています。

また、開発環境の \$BD/util/tool/src/\* にインストールされたネットワーク関連ツールのソースファイルも参考にしてください。

#### 注) 他マシンからの ping への応答について

TCP/IP マネージャがロードされていても、ネットワークを使用するプログラムが一つも存在しない状態では、ネットワークは停止状態になります。このとき、外部からの ping 要求などにも全く反応しませんので、ご注意ください。

ネットワークを動作状態にするには、ネットワークを使用するプログラムを起動するか、または、TCP/IP マネージャに対して明示的に so\_start() を発行する必要があります。

このためのユーティリティとして netstart があります。このプログラムを次のように動作させておけば、ネットワークは常に動作状態になり、外部からの ping 要求にも応答するようになります。

```
[/SYS]% netstart &
```

この netstart や so\_start() を実行しなくても、通常の手順でプログラムから TCP/IP 機

能を使い始めれば、その時点でネットワークは動作状態になります。ネットワークを使用するプログラムが、明示的に `so_start()` を発行する必要はありません。

## 4. ソフトウェア開発方法

### 4.1 開発環境のインストール

#### 4.1.1 Linux 環境で開発を行う場合

GNU 開発環境のインストールやコンパイル方法に関しては、

- ・T-Kernel 2.0 リファレンスキット GNU開発環境(Linux版)説明書

を参照してください。その後、後述する「PMC T-Shell 開発環境のインストール」を参照してください。

#### 4.1.2 Windows 上のコマンドライン環境で開発を行う場合

GNU 開発環境のインストールやコンパイル方法に関しては、

- ・Cygwin インストール方法説明書
- ・Cygwin 用 T-Kernel 2.0 リファレンスキット 開発環境インストール方法説明書
- ・T-Kernel 2.0 リファレンスキット GNU開発環境(Windows版)説明書

を参照してください。その後、後述する「PMC T-Shell 開発環境のインストール」を参照してください。

#### 4.1.3 Windows 上の GUI 環境で開発を行う場合

GNU 開発環境のインストールやコンパイル方法に関しては、

- ・Cygwin インストール方法説明書
- ・Eclipse インストール方法説明書
- ・Eclipse 用 T-Kernel 開発環境インストール方法説明書
- ・GNU開発環境(Eclipse版)説明書

を参照してください。開発に関する詳細は

- ・T-Kernel 2.0 リファレンスキット GNU開発環境(Windows版)説明書

を参考にしてください。

その後、後述する「PMC T-Shell 開発環境のインストール」を参照してください。

#### 4.1.4 PMC T-Shell 開発環境のインストール

##### (1) Linux 環境で開発する場合

PMC T-Shell を使用する上で必要なツール、ライブラリ、サンプルプログラム等を追加インストールするためには、本 CD-ROM 内の、`jp¥soft` ディレクトリに含まれている以下のファイルを開発環境のディレクトリ(\$BD)上で展開してください。

```
te.tshell.em1d512te.X.X.X.tar.gz
↑X は、バージョンを表す数字が入ります。
```

```
例 : % cd /usr/local/te
      % tar xzpf te.tshell.em1d512te.X.X.X.tar.gz
```

##### (2) Windows 上のコマンドライン環境で開発する場合

「(1) Linux 環境で開発する場合」と同じです。「(1) Linux 環境で開発する場合」を参照してください。

##### (3) Windows 上の GUI 環境で開発する場合

- ・PMC T-Shell (Eclipse版) インストール説明書

を参照してください。

##### (4) インストール内容

追加インストールされる内容は以下の通りです。

bapl/		
	dbox/*	データボックスの標準インクルードファイル ( <code>stddef.d</code> , <code>libapp.i</code> , <code>libapp.d</code> )
	tsh-sample/*	「PMC T-Shell プログラミング解説書」で参照している サンプルプログラム群
lib/		
	em1d512/	
	libtf.a	多国語(多漢字)処理ライブラリ
	libtf.so	(../em1d512.dl/libtf.so へのシンボリックリンク)
	em1d512.dl/	
	libtf.so.2	多国語(多漢字)処理ライブラリ(共有ライブラリ)
	libtf.so	(libtf.so.2 へのシンボリックリンク)
	libtf.a	
util/		
	tool/src/	ネットワーク関連ツールソースファイル
	Makefile.net	ネットワーク関連ツール用 Makefile
	fget.C	ftp によるファイルのダウン/アップロード
	netconf.C	NETCONF 作成ツール

netstart.c	ネットワーク起動ツール
ping.C	簡易 ping ツール
ppp.C	PPP 接続ツール
tcptest.C	TCP テスト用
telnet.c	簡易 telnet ツール
udptest.C	UDP テスト用

† データボックスコンパイラに関しては、開発環境に含まれるソースファイル (\$ (BD)/tool/build/databox/src/databox.C) 内の先頭にある説明を参照してください。

## 4.2 開発対象ソフトウェアの分類

開発対象となるソフトウェアは、大きく以下の 3 つに分類され、開発方法やプログラムのオブジェクト形式などが異なってきます。

T-Kernel 2.0 リファレンスキット では、プロセスベースのソフトウェアの開発が主体になります。

### ○ モニタベースのソフトウェア

T-Kernel の機能を使用しない、ハードウェア上で直接動作するソフトウェアで、T-Monitor によりプログラムをメモリ上にロードして実行します。

### ○ T-Kernel ベースのソフトウェア

T-Kernel の機能を使用するデバイスドライバやサブシステムなどのリアルタイムソフトウェアで、常駐メモリ上で動作します。

### ○ プロセスベースのソフトウェア

PMC T-Kernel Extension、および PMC T-Shell 上で、プロセスとして仮想メモリ上で動作するソフトウェアで、T-Kernel の機能を直接利用することはできません。一般のアプリケーションソフトウェアや、ライブラリ相当のミドルウェアなどが相当します。

#### 4.2.1 モニタベースのソフトウェア

モニタベースのソフトウェアは、MMU を使用しない環境で動作しますので、T-Monitor で使用している資源を除いてすべてのハードウェア資源を利用することができます。

実装仕様書を参照して、T-Monitor 自身が使用している RAM 領域を除いたアドレスにプログラムをリンクするようにしてください。

作成したプログラムは、S フォーマット形式に変換して、T-Monitor の Load コマンドによりメモリ上にロードして、Go コマンドで実行します。

メモリ上にロードしたプログラムは、T-Monitor の WriteDisk コマンドで作業用ディスクに保存し、ReadDisk コマンドでロードすることができます。

#### 4.2.2 T-Kernel ベースのソフトウェア

デバイスドライバなどの、T-Kernel ベースのソフトウェアは、MMU を使用した環境で、システムの共有空間に常駐して動作します。

CLI の `recv` コマンドを使用して、作業用ディスクに一旦ファイルとして格納してから、CLI または IMS の `lodspg` コマンドでロードおよび実行を開始します。`lodspg` を終了してもメモリ領域を占有したままとなります。プログラムの終了は `unlspg` コマンドで行い、プログラム終了後にメモリ領域は解放されます。

ロード時には、自動的にリロケーションが行われます。実際にロードしたアドレスは、`lodspg` コマンドで表示されますが、CLI の `ref spg` コマンドで確認することができます。

#### 4.2.3 プロセスベースのソフトウェア

一般アプリケーションとしてのプロセスベースのソフトウェアは、MMU を使用した環境で、ローカル空間にロードされて動作します。

CLI の `recv` コマンドを使用して、作業用ディスクに一旦ファイルとして格納してから、CLI または IMS の プログラム実行コマンドでロードおよび実行を開始します。

プロセスベースのソフトウェアはユーザレベルの保護レベルで動作しますので、T-Kernel の機能を直接利用することはできません。また、I/O 空間を直接アクセスすることはできません。

プロセスベースのソフトウェアでは、共有ライブラリを使用することができます。共有ライブラリ自体は、システム起動ディスクの `/SYS/lib/*` に含まれています。共有ライブラリを使用したプログラムの開発方法に関しては「T-Kernel 2.0 リファレンスキットGNU開発環境(Linux版)説明書」または「T-Kernel 2.0 リファレンスキットGNU開発環境(Windows版)説明書」を参照してください。

### 4.3 プログラム開発方法

実際の開発方法は、「T-Kernel 2.0 リファレンスキットGNU開発環境(Linux版)説明書」または「T-Kernel 2.0 リファレンスキットGNU開発環境(Windows版)説明書」を参照してください。

PMC T-Kernel Extension、PMC T-Shell を利用できるソフトウェアは、基本的にプロセスベースのソフトウェアとなりますので、開発環境のディレクトリ `$BD/bappl` で作成してください。

PMC T-Shell で提供している API に関しては、「PMC T-Shell説明書」および「PMC T-Shell プログラミング解説書」と、開発環境の `$BD/bappl/tsh-sample/*` にインストールされるサンプルプログラム群を参照してください。

マイクروسクリプトを使用したプログラムは、T-Kernel 2.0 リファレンスキット上で直接作成することもできますが、超漢字V(別売)上で作成し、作成した結果をT-Kernel 2.0 リファレンスキット上にコピーして実行することもできます。

マイクروسクリプトに関しては、超漢字V(別売)の電子マニュアル、または書籍を参照してください。

超漢字V (別売) で作成したマイクروسクリプトの実体は、USB メモリなどを利用してT-Kernel 2.0 リファレンスキットへコピー (実体複製) してください。コピーの方法などは、超漢字V (別売) の説明書をご参照ください。

#### 4.4 デバッグモード

開発用ホストマシンのターミナルソフト、または PMC T-Shell 上のコンソール小物から、以下のコマンドを使用してデバッグモードの切り替えができます。デバッグモードの切り替えは、再起動後に有効になります。

```
[/SYS]% debugmode 0    -- デバッグモードとしない  
[/SYS]% debugmode 1    -- デバッグモードとする
```

インストール直後は、デバッグモードになっています。

- ・デバッグモードの場合  
開発用ホストマシンのターミナルソフトからの操作ができます。  
システム終了は、ターミナルソフトからのコマンドにより行います。
- ・デバッグモードでない場合  
開発用ホストマシンのターミナルソフトからの操作はできません。(シリアルポートは使用されません)  
初期ウィンドウを閉じる (メニューから終了を実行する) と、自動的にシステム終了となります。

#### 4.5 ROM ディスクの更新

microSD カードまたは eMMC 上に作成した起動ディスクの内容全体を、以下の手順で、フラッシュメモリ上の ROM ディスクに書き込んで、ROM ディスクの内容を更新することができます。

- (1) microSD カードまたは eMMC に、ROM ディスクの最大サイズ (16580 KB) 以下のサイズの区画の起動ディスクを作成してください。  
この起動ディスクにファイルの追加/更新などを行い、ROM ディスクに書き込む内容を作成してください。また、実際に書き込む前に、この起動ディスクからシステムが正しく起動することを確認してください。

- (2) DIPSW-4 を ON にして、システムを起動して T-Monitor を立ち上げてください。

† SW-1 を押しながら、リセットスイッチを押しても T-Monitor が立ち上がります。

- (3) WriteRda (WRDA) コマンドを入力してください。

```
TM> WriteRda pcb0 (microSDカードの先頭区画の場合)  
pcc0 (eMMC先頭区画の場合)
```

- (4) 上記の操作により microSD カードまたは eMMC の区画の内容が、フラッシュメモリ上の ROM ディスクに書き込まれます。その後は、必ずシステムを再起動して動作を確認してください。

#### 4.6 T-Kernel 2.0 対応について

データタイプは原則として T-Kernel 1.0 に準じて定義されています。詳細は `include/typedef.h` を参照してください。

T-Kernel 2.0 の仕様書とライブラリ説明書には一部重複があります。重複部分に相違がある場合、ライブラリ説明書の方を参照してください。