

T-Kernel/Standard Extension ソースコード活用ノウハウのご紹介

ITRON仕様検討グループ T-Kernel/SE適応化WG

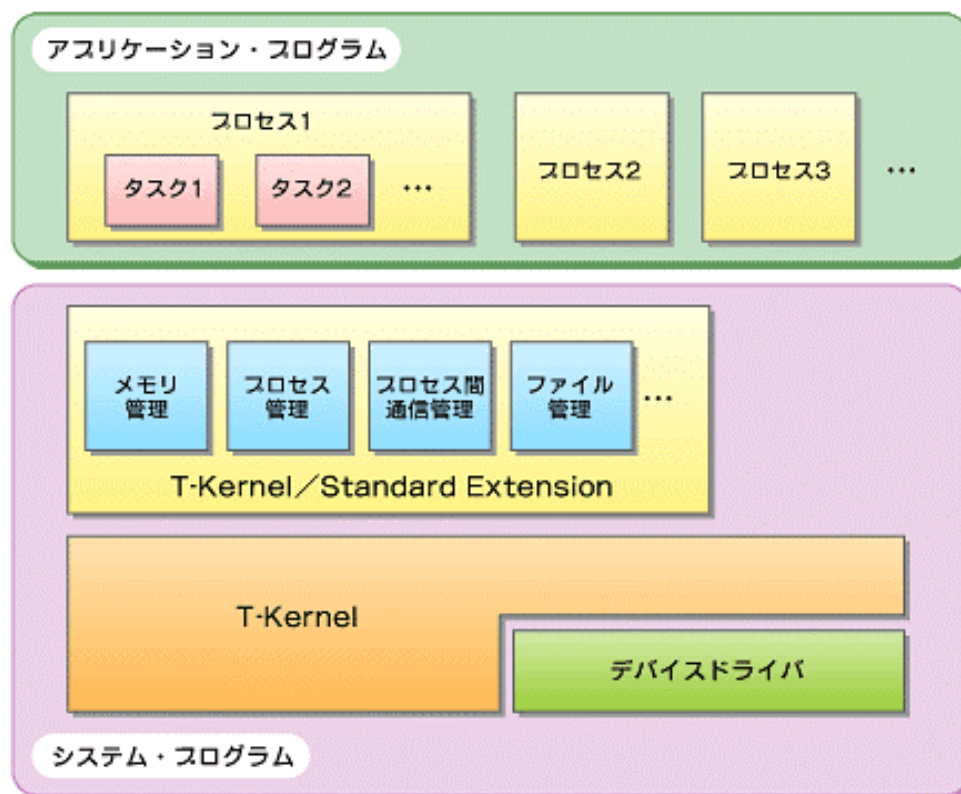
社団法人トロン協会 T-Kernel/SE適応化WG
2006年12月6日 TRONSHOW2007

目次

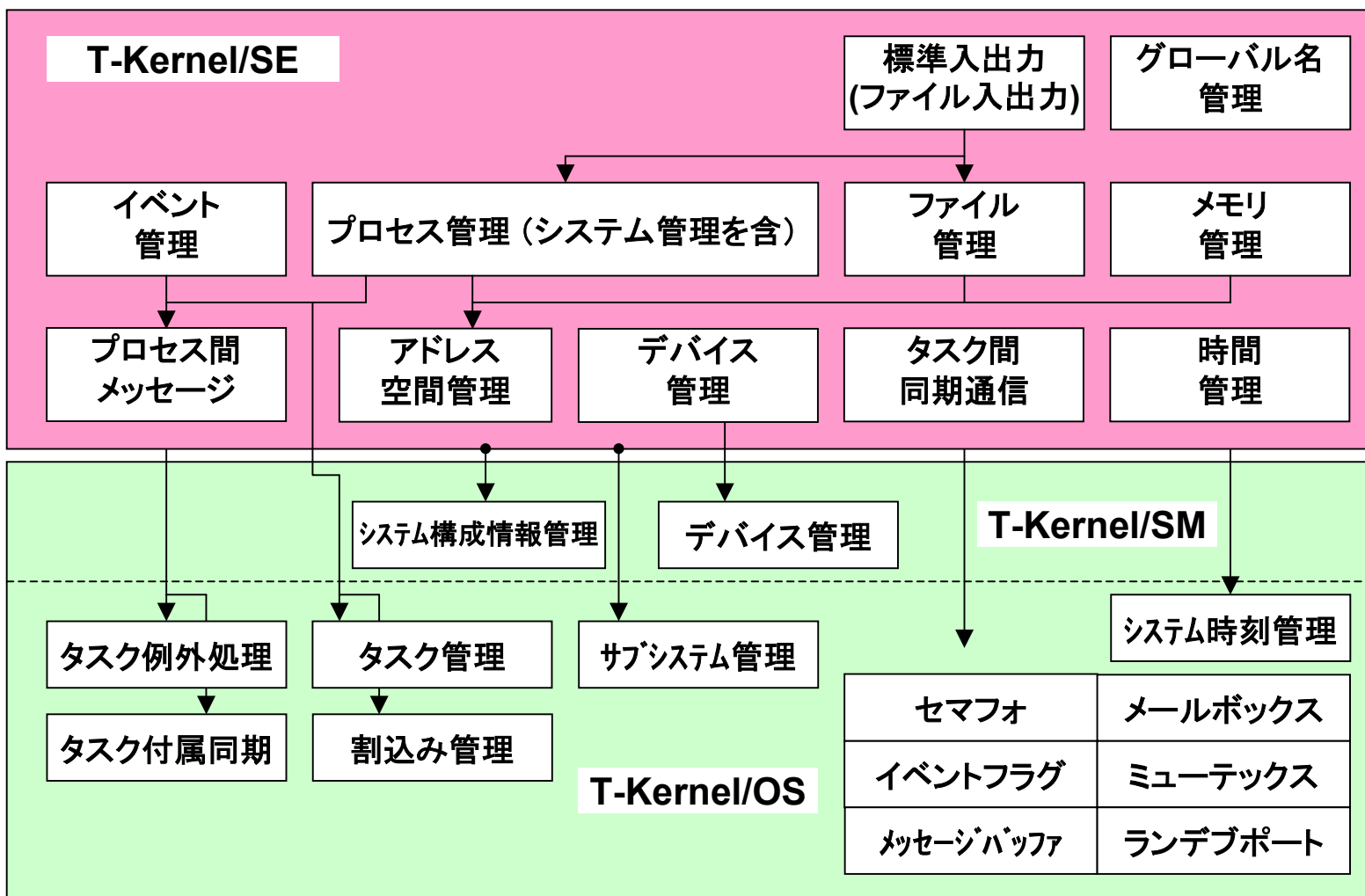
- T-Kernel/Standard Extensionとは
- 活動のねらい
- ワーキンググループメンバー
- 活動のご紹介
 - すぐ使えるT-Kernel/Standard Extension
 - ARM純正環境への適応
 - MIPSマイコン対応品種追加、TCP/IPの移植
 - MMU非搭載CPU(FR、V850)への適応

T-Kernel/Standard Extensionとは

- T-Kernelの機能を拡張し、より高度なOS機能を実現するプログラム



モジュール構成



活動のねらい

■ T-Kernel/SEの利用事例の情報発信

- 各種マイコンへ適用する際の検討事項を洗い出し、試行し、収集した情報をT-Engineフォーラムへフィードバック
- 適応事例を、セミナーやWebなどにより情報発信

利用方法解説

対応マイコン展開

デバイスドライバ適応

ワーキンググループメンバー

- ITRON仕様検討グループの有志
 - － イーソル(株)
 - － 富士通(株)
 - － 富士通デバイス(株)
 - － 三菱電機(株)
 - － NECエレクトロニクス(株)
 - － NEC東芝スペースシステム(株)
 - － (株)ルネサスソリューションズ
 - － (株)ルネサステクノロジ
 - － 横河デジタルコンピュータ(株)
 - － YRPユビキタス・ネットワーキング研究所

事例紹介

- すぐ使えるT-Kernel/Standard Extension
- ARM純正環境への適応
- MIPSマイコン対応品種追加、TCP/IPの移植
- MMU非搭載CPU (FR、V850)への適応

すぐ使えるT-Kernel/Standard Extension ～ソースコードの入手からサンプルアプリの実行まで～



岡崎 太
株式会社ルネサス ソリューションズ

社団法人トロン協会 T-Kernel/SE適応化WG
2006年12月6日 TRONSHOW2007

目次

1. T-Kernel/Standard Extensionソースコードを入手しよう
2. 開発環境を整えよう
3. ビルドしてみよう
4. サンプルアプリケーションを動かしてみよう
5. まとめ
6. 補足事項
7. 参考ドキュメント
8. 関連サイト

T-Kernel/Standard Extension

ソースコードを入手しよう

■ T-Engineフォーラムから一般公開中のソースコード

- T-Kernelソースコード v1.02.02
- T-Kernel/Standard Extensionソースコード v1.00.00
- 拡張ファイルシステム(FAT)ソースコード v1.00.00
- 拡張ファイルシステム(CD-ROM)ソースコード v1.00.00
- T-Kernel/Standard Extension
サンプルドライバソースコード v1.00.00
- T-Kernel/Standard Extension
サンプルアプリケーションソースコード v1.00.00

(06.12.01現在)

ダウンロードの方法

■ T-Engineフォーラムのサイトからダウンロード

URL: <http://www.t-engine.org/>

TOPページの左のメニューから“ダウンロード”を選ぶ。



ダウンロードの方法(2)

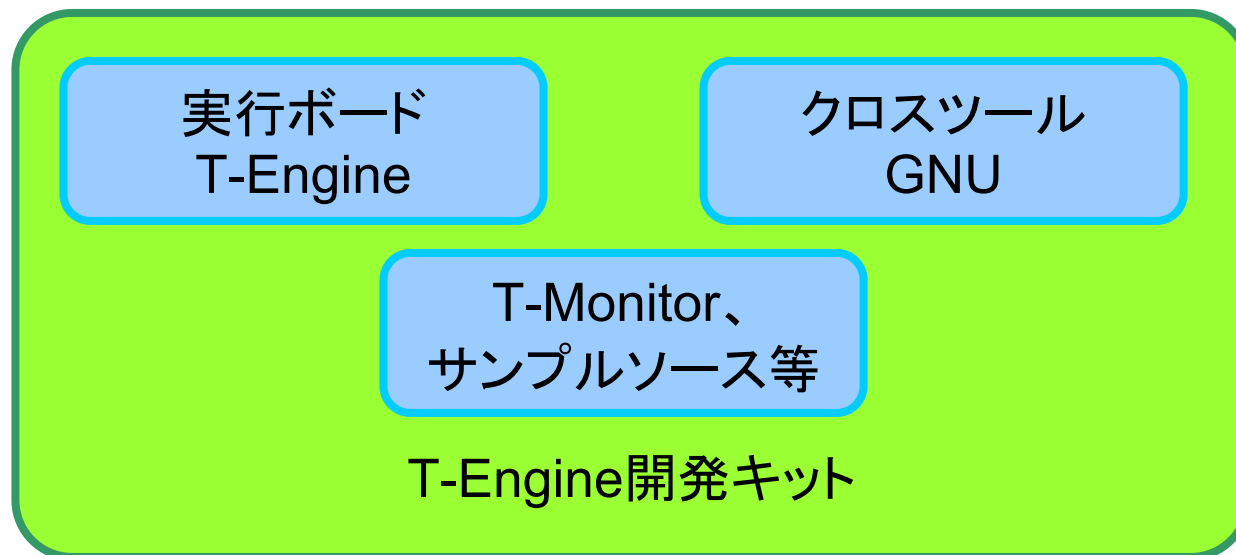
■ 利用申し込みとダウンロード

- ・T-License(ソースコード利用規約)等に承諾し、申し込みフォームに必要事項を入力する。
- ・T-Engineフォーラムより、ダウンロードページに入るためのIDとパスワードがメールで送られてくる。
- ・IDとパスワードを使ってダウンロードする。

The screenshot shows a web page titled "ダウンロード" (Download) in a red header. Below the header, there is a list of links: "T-Kernel ソースコード", "[T-Kernel 利用申込み](#)", and "[T-Kernel Version 1.02.02 ソースコードダウンロード](#)". A green arrow labeled "①利用申し込み" points to the "T-Kernel 利用申込み" link. Another green arrow labeled "②ダウンロード" points to the "T-Kernel Version 1.02.02 ソースコードダウンロード" link.

開発環境を準備する

- T-Kernelの標準開発プラットフォームT-Engine
 - ・ソースコードからカーネルを構築するためのクロスツールと、動かすための実行ボードが必要。



※開発キットによっては、製品構成が異なります。

※T-Monitorはオブジェクトが付属されています。

T-Engine開発キットを入手するには？

- SH系 (SH7727, SH7751R)
株式会社ルネサス テクノロジ
株式会社日立超LSIシステムズ
- ARM系 (ARM920, ARM720)
横河デジタルコンピュータ株式会社
- 上記、及びVR系 (VR5500, VR4131)
パーソナルメディア株式会社

※順不同

※T-Kernel/Standard Extensionが対応しているCPUのみ記載

開発環境をセットアップする

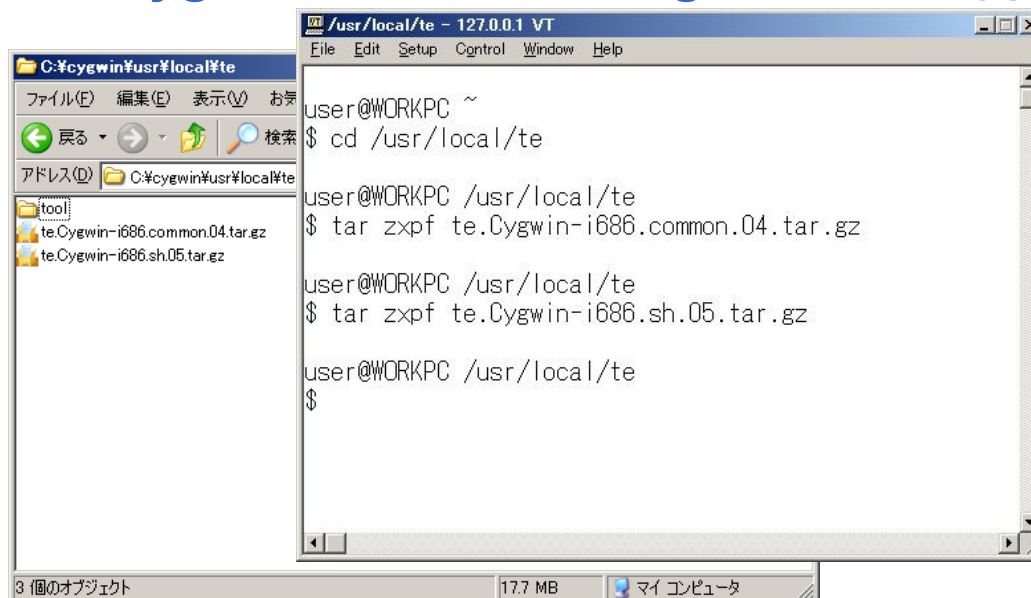
■ 開発PC

- Windows又はLinuxで開発
 - ・開発キットにはWindows用とLinux用の2つのGNUツールが付属している。
- Windowsで開発する場合
 - ・Windows上でGNUツールを使えるようにするソフトウェア“**Cygwin**”をインストールする。

GNUツールをセットアップする

- CD-ROMに付属のGNUツールを展開する
/usr/local/teに展開する。

- ・te.Cygwin-i686.common.04.tar.gz Cygwin 用プラットフォーム共通部分
- ・te.Cygwin-i686.sh.05.tar.gz Cygwin 用SH7727 対応部分



展開すると“tool”という
ディレクトリができる

Cygwinの環境変数を設定する

- 作業ディレクトリやGNUツールディレクトリのパスを指定する
 - ・HOMEディレクトリ下にある.bashrcに設定する。

```
# Some example functions
# function settitle() { echo -ne "%e]2;${@%a}%e]1;${@%a}"; }

PATH=$PATH:~/bin

export BD=/usr/local/te
export GNUs=/usr
export GNU_BD=$BD/tool/Cygwin-i686
export GNUsh=$GNU_BD/sh-unknown-tkernel
export GCC_EXEC_PREFIX=$GNU_BD/lib/gcc-lib/
```

※設定値は、標準インストールの場合

ソースファイルを展開する

- ダウンロードしたファイルを解凍する
 - ・ダウンロードしたファイルは、TAR形式のアーカイブファイルなので、Cygwinのtarコマンドで適当なディレクトリに解凍する。
- 環境変数BDに設定したディレクトリに展開する
 - ・標準インストールの場合、**/usr/local/te**に展開する。
 - ・展開する順番は以下の通り。
 1. T-Kernel
 2. T-Kernel/Standard Extension
 3. サンプルドライバ、拡張ファイルシステム、サンプルアプリ
(順番関係なし)

ソースファイルをビルドする

- ドライバやライブラリを順にmakeする
 - RomInfo(カーネルコンフィグレーション情報)
 - ・ロードオブジェクト**rominfo.mot**が生成される。
 - T-Kernelライブラリ
 - T-Kernel/Standard Extensionライブラリ
 - ・共有ライブラリ**libxxxx.so.x**が生成される。
 - デバイスドライバI/Fライブラリ
 - サンプルデバイスドライバ
 - カーネル本体
 - ・ロードオブジェクト**kernel_rom.mot**が生成される。
 - ユーザアプリケーション
 - ・各サンプルアプリケーションなど

T-Engineに書き込む

■ 書き込むもの

- ・以下のmotファイルをT-EngineのFLASHメモリに書き込む。
デバッグボード上のEPROMに入っている、書き込みプログラムを使用する。

- RomInfo : **rominfo.mot**

te/config/tkse/build/std_xxxx/rominfo.mot

- カーネル本体 : **kernel_rom.mot**

(T-Kernel、T-Kernel/SE、ドライバ等を一つにまとめた物)

te/tkse/build/std_xxxx/kernel_rom.mot

- T-Monitor : **tmonitor.mot**

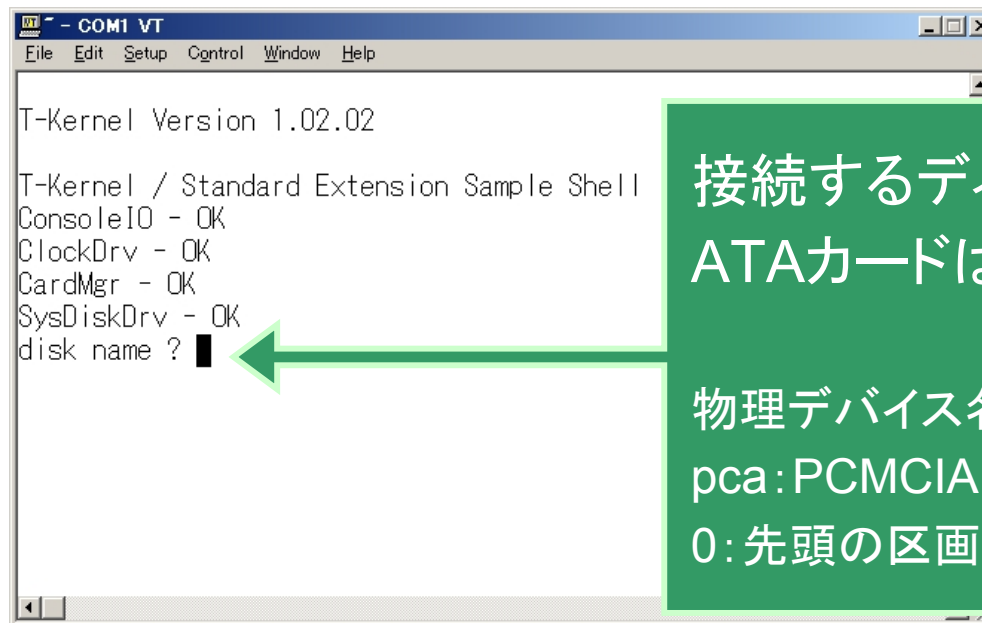
T-Engineには転送済み(開発キットに付属)のため、転送は**不要**。

アプリケーションの準備

- ディスクにはATAカードを使用する
 - ・扱いやすいFATでフォーマット。
- アプリケーションの実行オブジェクトをディスクに保存する
 - ・アプリの実行オブジェクトや、DLLファイルはATAカード内のどこに置いてもOK。(パスを指定できるため)
 - ・共有ライブラリは、**/libディレクトリ**に入れる。
(T-Kernel/SEが自動的に読み込むため、置き場所は固定)
 - ・ユーザ作成ライブラリその他、TK/SEの共有ライブラリも/libに入れる。
 - libdl.so.2 : DLL I/Fライブラリ
 - libg.so.2 : TK/SE標準汎用ライブラリ(標準Cライブラリ)
 - libtctr.so.2 : TRON Code文字列操作ライブラリ
 - libtkse.so.2 : TK/SEライブラリ

T-Engineを起動

- T-Engineを起動
 - ・簡易Shellが起動する。



```
COM1 VT
File Edit Setup Control Window Help

T-Kernel Version 1.02.02

T-Kernel / Standard Extension Sample Shell
ConsoleIO - OK
ClockDrv - OK
CardMgr - OK
SysDiskDrv - OK
disk name ? █
```

接続するディスク名を入力
ATAカードは“pca0”

物理デバイス名＋サブユニット番号
pca:PCMCIA
0:先頭の区画(パーティション)

簡易Shellを使う

■ コマンドを試してみよう

load : ファイルのロード(ディスクへ転送)

ls : ディレクトリエントリの表示

mkdir : ディレクトリ作成

rmdir : ディレクトリ削除

rm : ディレクトリエントリの削除

cd : ディレクトリ移動

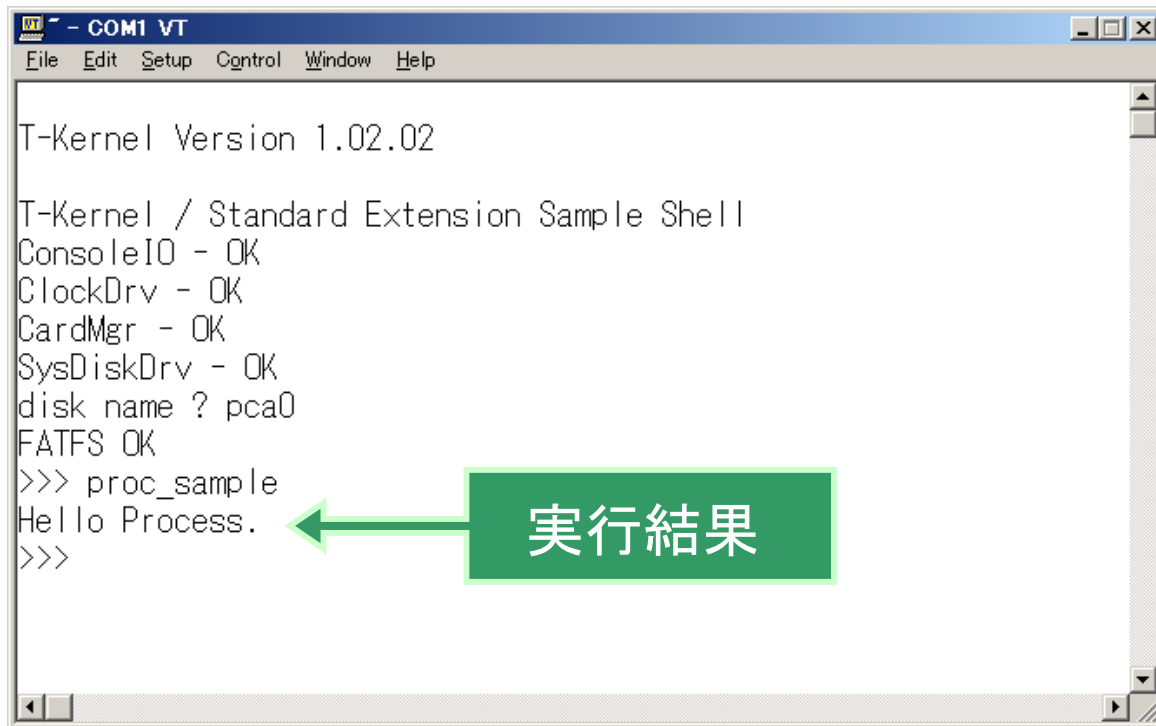
pwd : 現在のディレクトリ表示

? : コマンドヘルプ

アプリケーションの実行

■ アプリを起動

- ・簡易Shellのプロンプトに続けてファイル名を入力するだけ。



The screenshot shows a terminal window titled "COM1 VT" with a menu bar (File, Edit, Setup, Control, Window, Help). The text inside the terminal is as follows:

```
T-Kernel Version 1.02.02  
  
T-Kernel / Standard Extension Sample Shell  
ConsoleIO - OK  
ClockDrv - OK  
CardMgr - OK  
SysDiskDrv - OK  
disk name ? pca0  
FATFS OK  
>>> proc_sample  
Hello Process.  
>>>
```

A green arrow points from a green box labeled "実行結果" (Execution Result) to the output "Hello Process." in the terminal.

まとめ

■ T-Kernelはオープンソース

- ・カーネルからドライバまで全てオープンソース。
- ・インターネットに繋がっていれば、誰でも入手できます。

■ トロン協会ブースにてデモを展示してます

- ・フリーのスクリーンドライバ、KB/PDドライバ、フォントを追加したデモを展示しています。

補足事項

- T-Engine開発キットに付属のCD-ROMのバージョンは最新を使う
 - ・古いものは共有ライブラリ、DLLの構築ができないため。
 - ・詳しくは、パーソナルメディア株式会社のユーザサポートページを参照してください。
<http://www.t-engine4u.com/support/support.html>
- トロン文字コードを使用する場合
 - ・te.resource.sh.15.tar.gzに含まれるTCコード変換ツール“wch2hex”が必要。
 - ・使用方法は、同アーカイブの/etc/makerulesを参照してください。
- RAMディスクを使用する場合
 - ・今のStandard Extensionやドライバには論理フォーマットの機能がないため、ユーザで論理フォーマット処理を作ることによって利用可能。

補足事項(2)

- サンプルアプリケーション「sample_dll」と「sample_so」でリンクエラーが発生する
 - ・会員向けに公開されている「標準汎用ライブラリ(Cライブラリ相当)」を組み込むと改善します。本件について「サンプルアプリケーション説明書 v1.00.01」に補足が記載されているので、そちらを参照してください。
- サンプルドライバを組み込んだ場合、標準汎用ライブラリを組み込まないと、カーネル本体のmakeでリンクエラーが発生する
 - ・サンプルドライバのダウンロードページに対策方法が掲載されています。

参考ドキュメント

■ 開発環境、GNUツールについて

「GNU開発環境説明書」「GNU開発環境(Windows版)説明書」

入手先: T-Engine開発キットに付属

■ アーカイブの解凍とビルド方法について

「T-Kernelソースコード説明書」

「T-Kernel/Standard Extension構築説明書」

「T-Kernel/Standard Extensionサンプルデバイスドライバ構築説明書」

「T-Kernel/Standard Extensionサンプルアプリケーション構築説明書」

入手先: T-EngineフォーラムWebサイト

■ T-Engineについて(SH7727の場合)

SH7727 T-Engine開発キット概説書(ルネサス製開発キット)

SH7727 T-Engine Board概説書(日立超LSIシステムズ社製開発キット)

T-Engine/SH7727開発キット取扱説明書(パーソナルメディア社製開発キット)

入手先: 各社製T-Engine開発キットに付属

関連サイト

- SH/M32R T-Engine Home Page
<http://www.superh-tkernel.org/>

ARM純正環境への適応 ～移植のポイントについて～



内生藏 伸
横河デジタルコンピュータ株式会社

社団法人トロン協会 T-Kernel/SE適応化WG
2006年12月6日 TRONSHOW2007

目次

- はじめに
- 環境
- ARM920→ARM946移植ポイント
- GNU→RVCTアセンブラ展開
 - － アセンブラ対応: 変更例
 - － アセンブラ対応: インライン展開実例
 - － アセンブラ対応: 拡張SVCライブラリ
 - － アセンブラ対応: 拡張SVC実例
- 標準ライブラリについて
- まとめ
- FAQ

はじめに

- 待望のSE & ドライバーサンプル公開
 - T-Engineフォーラムより、すぐに動作させられるようにドライバー、サンプルプログラムが公開されました。
 - これにより、移植ベースとして動作確認が容易にとれる環境が整いました。
- GNU→ARM純正環境
 - 実際に簡単に扱えるものなのか、オープンソース対象外のT-Engineボード(ARM926)への移植、ならびにARM純正環境への対応を実施。
 - そこでの移植ポイントを説明します。

環境

- ターゲット
 - freescale社 MC9328MX21 (ARM926コア) 搭載T-Engineボード
- T-Kernel (T-Engineフォーラム公開)
 - T-Kernel 1.02.02
 - T-Kernel/SE 1.00.00
 - 拡張FAT ファイルシステム
 - サンプルドライバ、サンプルアプリ
- ◆ ARM920用 (std_mc9328) をベースに使用
- T-Monitor
 - 独自対応
- ARM純正環境
 - ARM社純正 RealView™ Compilation Tools (RVCT) Ver.2.2



RealViewはARM Limitedの商標です。

ARM920→ARM926移植ポイント

- ボード固有アドレス、固有設定値の変更
 - 開発キット:T-Engine/ARM926-MX21 の実装仕様書を参考に機種依存部を変更。
- CPU依存部として
 - 基本アーキテクチャが同一の為、特に変更はいりません。

GNU→RVCT アセンブラ展開

- アセンブラソース、インクルードファイルの変更
 - 表記法が違いますので、全面書き換えが必要となります。
 - 宣言、ラベル
 - 実行コードは、ほぼ流用可能。
- Cソース: インラインアセンブラの表記変更
 - 表記法が違いますので、全面書き換えが必要となります。
 - オープンソースを参考にする場合、インラインアセンブラ部はGNU拡張書式で記述されている部分がありますので、展開の必要性、あるいは関数化などの対応が必要となります。
- アセンブラソース用インクルードファイルの展開
 - GNU ASMでは、C言語の記述様式でインクルード可能なため、対応ファイルの作成、または変更が必要となります。
 - ssid.h → rvdsssid.h (ASM用)

	GNU		RVCT	
コメント	Cの書式	<code>/* */</code>		先頭行に';'
インクルード	Cの書式	<code>#include <machine.h></code>	GET	GET machine.h
デファイン	Cの書式	<code>#define PSR_I 0x00000080</code>	EQU	PSR_I EQU 0x00000080

```

T-Kernel
Copyright
T-Kernel

Version:
Released:

↓

0x00000000
Enable/disable
↓
#include <machine/
#include <tk/s

Disable external
UART interrupt
CPSR interrupt
Return value
↓
.text
.align 4
.global _start
.type _start, @function
↓
mrs r0, cpsr
orr ip, r0, #0x00000000
msr cpsr_c, r0
bx lr
↓

```

```

; /*
; * -----
; * T-Kernel↓
; *
; * Copyright (C) 2004-2006 by Ken Sakamura. All rights reserved.↓
; * T-Kernel is distributed under the T-License.↓
; * -----
; *
; * Version: 1.02.02↓
; * Released by T-Engine Forum(http://www.t-engine.org) at 2006/8/9.↓
; * -----
; */
;
; Arranged by Yokogawa Digital Computer Corporation ©2006↓
; */
↓
; /*
; * @(#)disint.S (libtk/MC9328MX21)↓
; *
; * Enable/disable external interrupt↓
; */
↓
PSR_I EQU 0x00000080↓
PSR_F EQU 0x00000040↓
PSR_DI EQU (PSR_I | PSR_F)↓
↓
EXPORT disint↓
EXPORT enaint↓
↓
AREA Int, CODE, READONLY↓
↓
; /*
; * Disable external interrupt↓
; * UINT disint( void )↓
; * CPSR interrupt flag is in disabled status.↓
; * Return value is CPSR prior to change.↓
; */
disint↓
> mrs r0, cpsr↓
> orr r12, r0, #PSR_DI↓
> msr cpsr_c, r12↓
> bx lr↓

```

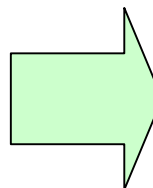
アセンブラ対応：インライン展開実例

```

/*↓
 * TLB-related↓
 */↓
↓
/*↓
 * Purge the whole TLB↓
 */↓
↓
{↓
> Asm("mcr p15, 0, %0, cr8, c7, 0:: "r"(0));↓
}↓
/*↓
 * Write back page tables to the main memory.↓
 */↓
↓
{↓
> Asm("mcr p15, 0, %0, cr7, c10, 1:: "r"((UW)ptbl & ~0x1f));↓
}↓

```

CPU.h



```

/*↓
 * TLB-related↓
 */↓
↓
/*↓
 * Purge the whole TLB↓
 */↓
↓
{↓
> UINT> reg = 0;↓
> asm↓
> {↓
> > mcr p15, 0, reg, c8, c7, 0↓
> }↓
}↓
/*↓
 * Write back page tables to the main memory.↓
 */↓
↓
{↓
> UINT> reg;↓
> reg = (UW)ptbl & ~0x1f;↓
> asm↓
> {↓
> > mcr p15, 0, reg, c7, c10, 1↓
> }↓
}↓

```

アセンブラ対応：拡張SVCライブラリ

- 拡張SVCライブラリとは？
 - T-Kernelでは、システムコールやドライバはソフトウェア割り込み(SWI)による起動となります。
 - 割り込みを行う部分のプログラムは、アセンブラ表記となります。
 - libsvc : 202ファイル
 - libsesvc : 215ファイル
- 拡張SVCのファイルは、make により自動生成(perl)される構成となっています。
 - サンプルドライバの拡張SVCは、make により自動生成。
- 拡張SVCライブラリのRVCT対応のアセンブラファイルも自動生成可能。
 - 今回は、力技で修正を行いました。
 - Perl等で作業の簡易化は可能。

```

T-Kerne
Copyright
T-Kerne
unde


Version
Releas

include <ma
include <tk
include <ex

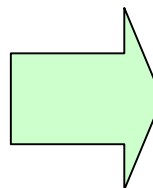
.text
.align
.globl
.type

stmfd
mov> r0,
stmfd
ldr> ip,
swi> SWI_
ldmfd
add> sp,
bx> lr

```



実行コードは一緒



新設:必要なもののみ定義

```

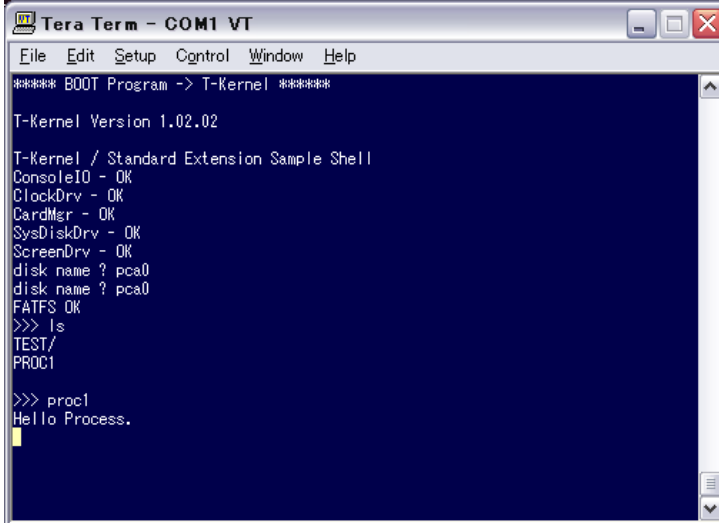
;
; libsvc_com.h
;
SWI_SVC      EQU 6
SWI_DEBUG    EQU 9

                        END

```

標準ライブラリの問題

- printf 等、処理系の標準ライブラリがリンクされてしまう。
 - 処理系の標準ライブラリをリンクさせない回避方法もあるようだが……（調査時間の問題から今後の課題）
 - 回避用に必要関数を作成。
 - printf2 / sprintf2 / vsprintf2



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
**** BOOT Program -> T-Kernel ****

T-Kernel Version 1.02.02

T-Kernel / Standard Extension Sample Shell
ConsoleIO - OK
ClockDrv - OK
CardMgr - OK
SysDiskDrv - OK
ScreenDrv - OK
disk name ? pca0
disk name ? pca0
FATFS OK
>>> ls
TEST/
PROC1

>>> proc1
Hello Process.
```


まとめ

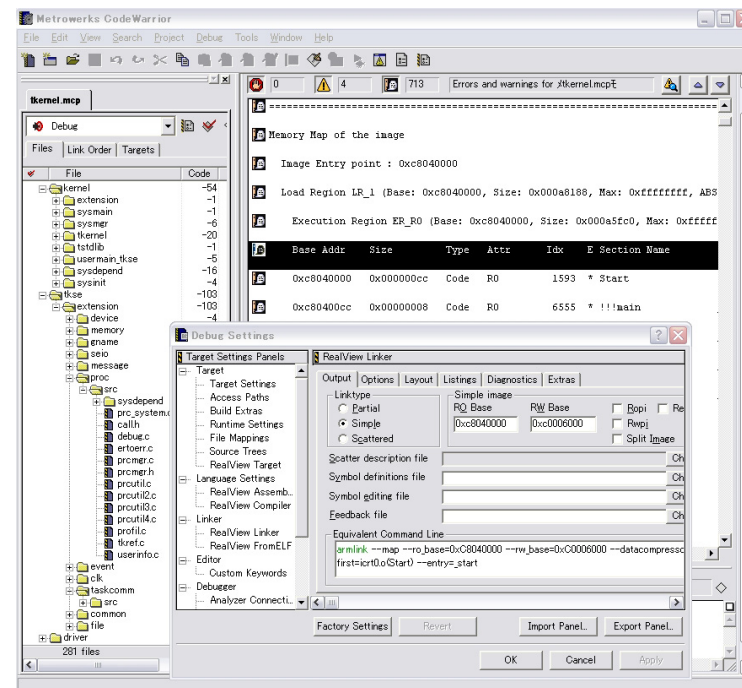
- 移植には、アセンブラファイルの展開という作業がありますが、変更用のスクリプトなど作成することで、回避可能です。
- その他は、提供ソースの大幅な変更なしに動作確認がとれました。
- OSを扱うには、意外と敷居が高いものですが、GNU以外でも、他の環境へ適応が可能ですので、気軽にT-Kernelをご利用してはいかがでしょうか。
- 今後
 - － テストスイートでの確認
 - － 拡張SVCファイルのRVCT用変換ツール(スクリプト)作成
 - － 標準ライブラリの依存調査

FAQ

■ プロジェクト構成は？

- ◆ T-Monitor
- ◆ T-Kernel (SEを含む構成)
- ◆ Config (RomInfo)
- ◆ 各ライブラリ

- 上記をプロジェクトを起こします。
- RVCT2.2 は、MetroWerks社の CodeWarrior IDE上で動作し、簡単にプロジェクト管理が行えます。



FAQ

■ コンパイラ等のオプション指定は？

➤ コンパイラ

- ・ `armcc --debug --cpu=ARM926EJ-S -O0 -DSTD_MC9328MX21 -D_STD_MC9328MX21_ -DUSE_UNICODE`

➤ アセンブラ

- ・ `armasm --debug --keep --cpu=ARM926EJ-S`

➤ リンカ

- ・ `armlink --map --ro_base=0xC8040000 --rw_base=0xC0006000 --datacompressor=off --first=icrt0.o(Start) --entry=_start`

FAQ

- `wchar_t` が重複定義でエラーとなりますが？
 - `include¥stddef.h` での定義を残し、
以下で定義されている箇所をコメントアウトします。
 - `include¥libstr.h`
 - `include¥tkse¥stdtype.h`
 - `kernel¥tstdlib¥bitop.h`

FAQ

- インクルードファイル中で、EXPORT宣言されている変数があり、複数ソースでインクルードされるため重複定義としてエラーとなりますが？
 - GCCにおいては1シンボル(領域)として解釈されるのでエラーにはなりません。
 - 以下に示すように対処します。
- ◆ T-Engineフォーラムでの次回改修予定に入っているそうです。

【変数】

EXPORT DE *DiskEntryTable;

【ファイル】

¥tkse¥extension¥memory¥src¥

segmgr. h

segio. c

【対処例】

segmgr. h 内記述をComment Out

【変数】

EXPORT FastLock BDevMgrLock;

【ファイル】

¥tkse¥extension¥device¥src¥

devmgr. h

【対処例】

main. c 内で EXPORT宣言

devmgr. h では、IMPORT宣言に変更。

FAQ

■ longlong の対応は？

- GNU同様の扱いができます。
- ソース内では、`#ifdef __GNU__` で切り分けられていますので、対応できるように変更します。一時的に `#define __GNU__` とし、定義後、`#undef` して対応をとります。
 - `kernel¥kernel¥src¥longlong.h`
 - `include¥tkse¥longlong.h`
 - `include¥tkse¥seio¥machine¥types.h`

FAQ

■ デバッグ方法は？

- ICE (インサーキットエミュレータ) を使用します。
- ボードへの移植、割り込み、カーネル動作を追うには必需です。
- 弊社 advicePRO をお勧めします。
 - オープンソース版T-Kernel対応デバッガを開発中！
 - システムプログラム、プロセスデバッグ対応



■ ARM純正環境の入手は？

- ARM開発ツール 正式ディストリビュータである弊社までお問い合わせ下さい。
- 英ARM社製ツールの販売、サポート、トレーニング等をご提供しております。

MIPSマイコン対応品種追加、TCP/IPの移植

-MIPS T-Engine/VR5701への移植-



松井 研司
NECエレクトロニクス株式会社

社団法人トロン協会 T-Kernel/SE適応化WG
2006年12月6日 TRONSHOW2007

目次

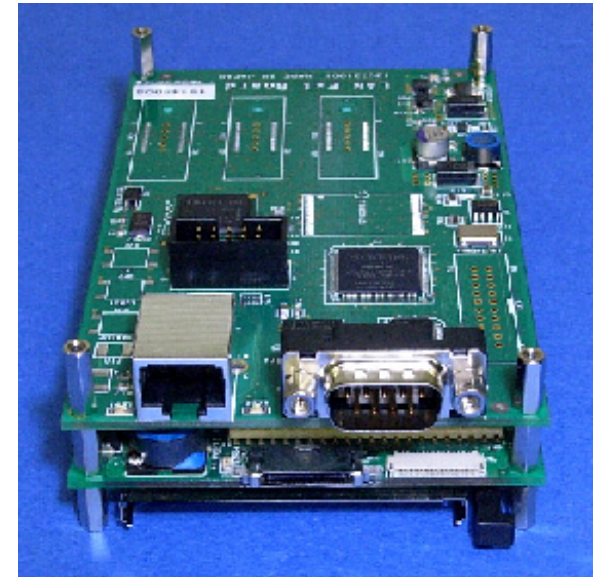
- 目的
- 環境
- 対象マイコン、ボードの違いの洗い出し
- T-Monitorの移植
- T-Kernelの移植
- T-Kernel/SEの移植
- FATファイルシステムの組み込み
- TCP/IPプロトコルスタックの移植
- 移植結果
- まとめ
- FAQ

目的

- T-EngineフォーラムでサポートされているVR5500の後継マイコンへの適応
- T-Monitorを整備し、オープンソフト環境の充実
- T-Kernelで動作するTCP/IPプロトコルスタックのT-Kernel/SEへの移行

環境

- ターゲット
 - パーソナルメディア社の T-Engine/VR5701開発キット およびT-Engine/VR, TX用拡張LANボード
- T-Kernel(T-Engineフォーラム公開)
 - T-Kernel 1.02.02
 - T-Kernel/SE 1.00.00
 - 拡張FAT ファイルシステム
 - サンプルドライバ ソースコード
 - サンプルアプリ、ソースコード
 - ◆ VR5500用をベースに使用
- T-Monitor
 - NECソフト様開発のものを利用
- コンパイラ
 - パーソナルメディア社の開発キット添付のGNUコンパイラ gcc V3.3.2
- エミュレータ(フラッシュメモリ書き込み)
 - 京都マイクロコンピュータ社製 PARTNER-Jet



T-Engine/VR5500とT-Engine/VR5701 の違いの洗い出し

1. PCI空間
2. 割込みコントローラ
3. GPIO
4. SDRAM BANK処理

※VR5500:SDRAM、VR5701:DDR-SDRAM

6. PCMCIA

※VR5500:R5C475 II、VR5701:R5C485

マイコン コアの
アーキテクチャは
同一

T-Monitorの移植

- 周辺I/O初期化部分の作りこみに留意
- システム機能
 - － ハードウェア初期化
 - － システムの起動
 - － 例外/割込み/トラップ処理機能
- デバッグ機能
- プログラムサポート機能

NECソフト株式会社
様作成のVR5500用
T-Monitorを改造

T-Kernelの移植

- GPIO/PCI/SDRAMなどの初期化部分をモニタの初期化部と整合

周辺I/Oの違いに
対応する部分の
作りこみ

- ライブラリ、カーネル、設定ファイルの構築
- T-Monitorのターゲットボードへの書き込み
- T-Kernelのターゲットボードへの書き込み

T-Kernel/SEの移植

- T-Kernel/SEの処理部分では、VR5500とVR5701で変更すべき点はなし
- コンパイルして、ターゲットへ書き込み
 - T-Kernelライブラリの構築
 - T-Kernel/SEライブラリの構築
 - T-Kernel/SEの構築
 - 設定ファイルの構築

Standard Extensionは
CPUアーキテクチャが
同じであれば、
変更する部分はない

FATファイルシステムの組み込み

- FATファイルシステムの処理部分では、VR5500とVR5701で変更すべき点はなし
- PCMCIAスロットを介して、コンパクトフラッシュをアクセスするためのドライバの移植
 - PCMCIAデバイスの差異の確認
 - VR5500:R5C475 II、VR5701:R5C485
 - 割込みコントローラ部の変更
 - T-Monitorの初期化部と矛盾がないことの確認

※FATファイルシステムは使用に際しての制限があります。
詳細についてはT-Kernel/SEソースコード公開ページにある
「T-Kernel/Standard Extension 構築説明書」を参照してください。

TCP/IPプロトコルスタックの移植

-ミドルウェアの入手-

- SH/M32R T-Engine Home Page
<http://www.superh-tkernel.org/jpn/> に掲載されているT-Kernel用TCP/IPプロトコルスタックをダウンロード
- LANデバイス(Intel 82551ER)用のデバイスドライバの作成

<http://www.superh-tkernel.org/jpn/>には
実践的な参考情報が
いろいろ載っておりますので
ぜひご参照ください

TCP/IPプロトコルスタックの移植

-LANDドライバの作成-

- LANDドライバの初期設定とT-Monitorとの整合を確認
 - － 割込みコントローラ
 - ・ 割込みルーティング、割込みマスク設定
 - － 外部PCI
 - ・ 割込みマスク設定
- MMUを用いたメモリ管理への対応
 - ・ CPUから見えるアドレスと、LANDドライバから見えるアドレスの整合
- T-KernelのsearchPciDev()機能に、T-Monitorが対応していることの確認

参考文献
TRONWARE Vol.99

TCP/IPプロトコルスタックの移植

-プロトコルスタック処理内での対応-

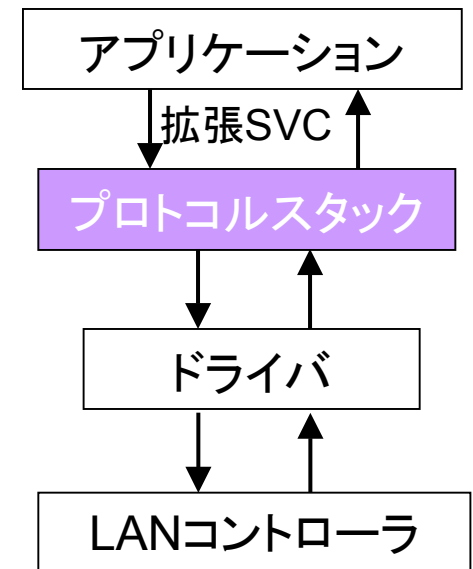
- パックした構造体に対応するコンパイルオプション調整
- エンディアンの違いの対応
- 割込み禁止、許可のタイミング

T-Engineフォーラムから、
各種マイコンにポーティング済みの
プロトコルスタックが
リリースされる
ことを期待します

TCP/IPプロトコルスタックの移植

-T-Kernel/SEへ組み込み-

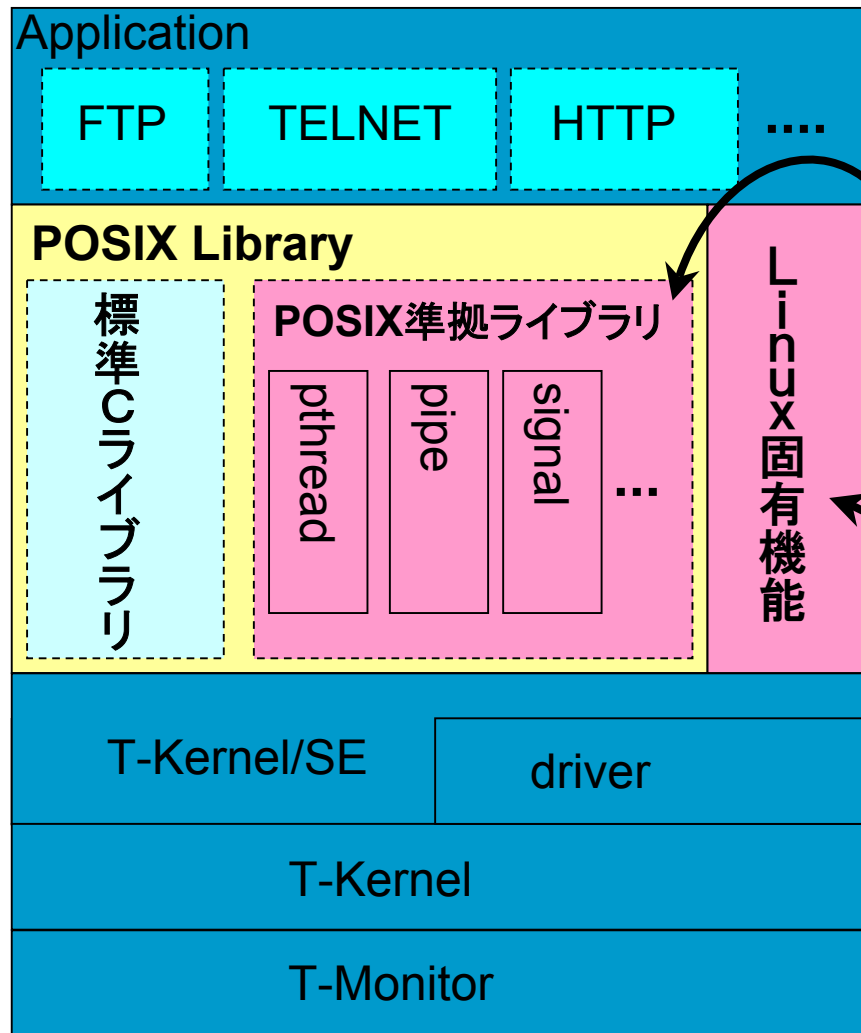
- ミドルウェア部はサブシステムとして登録
- T-Kernelで動作確認したミドルウェア、ドライバは、T-Kernel/SE上でも容易に動作可能



移植結果

- T-Monitor
 - サイズ:113KB
- T-Kernel + T-Kernel Standard Extension
 - サイズ:592KB
 - 標準Cライブラリ、ファイルシステム(FAT、CD-ROM)、一般公開版サンプルドライバを含む
- TCP/IP
 - TCP/IP機能サブシステム サイズ:144KB

POSIXライブラリの整備



NECソフト株式会社のとりくみ

- FTPやTELNETなどのLinux等で活用しているオープンソースソフトウェアを、簡易にT-Engine上に移植する目的で、POSIX準拠ライブラリの整備中
- 同時にLinux固有機能(ユーザ管理など)のラッピング関数も整備

まとめ

- MIPSアーキテクチャのマイコン間での移植はきわめて容易
- デバイスドライバ作成にあたり、初期化処理はT-Monitorの初期化部との整合に留意必要
- 今回の成果のT-KernelのVR5701パッチ、T-Monitorは今後公開予定

FAQ

- T-Kernelをコンパイルしたところ、次のエラーが発生しましたが？
/usr/local/te/tool/Linux-i686/ mips64el-unknown-tkernel/bin/gccvr :コマンドが見つかりませんでした。
 - gccのバージョンの違いによるエラーで
「T-Kernel ソースコード説明書 Rev.1.02.02」
「7. バージョンの異なる GNU 開発環境への対応」
を参照してください

GNUvr を GNUmips へ変更

```
export GNUvr=$GNU_BD/mips64el-unknown-tkernel
```

↓

```
export GNUmips=$GNU_BD/mips64el-unknown-tkernel
```


MMU非搭載CPU (FR、V850) への適応



小島 史郎
富士通デバイス株式会社

高倉 規彰 樋口 正雄
NECエレクトロニクス株式会社

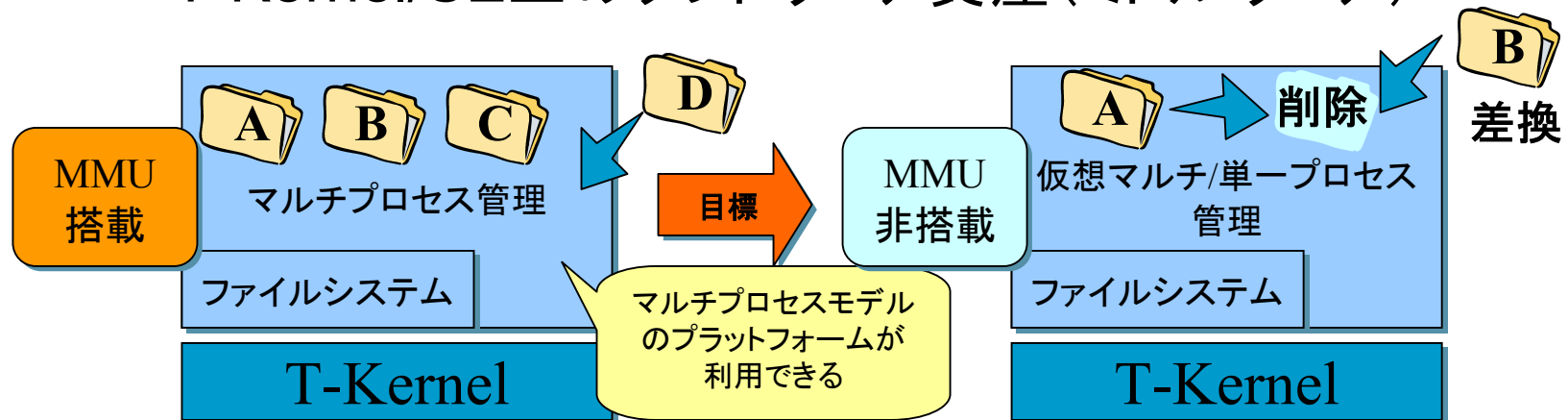
社団法人トロン協会 T-Kernel/SE適応化WG
2006年12月6日 TRONSHOW2007

目次

- MMU非搭載CPU適応化の動機
- 対象CPU
- 対象CPUへ適応する場合の問題点
- MMU非搭載CPU向け適応仕様
- 適応化実装
- 適応化結果
- デモ内容
- まとめ

MMU非搭載CPU適応化の動機

- T-Kernel/SEの恩恵をMMU非搭載CPUでも！
 - T-Kernel/SEのもつ機能(ファイルシステム)
 - T-Kernel/SE上のソフトウェア資産(ミドルウェア)



マルチプロセス(→多機能)は性能的に困難かもしれないが、
単機能のアプライアンスには適用可能なのでは？

対象CPU

■ FRファミリ(富士通)

- T-Engine Appliance Ansel-Tea/FR MB91403
(富士通デバイス製)
 - 32ビットRISC CPU「MB91403」 動作周波数 33MHz
 - CPU内蔵メモリ RAM 64KB
 - ボード搭載メモリ FLASH-ROM 2MB / SDRAM 2MB

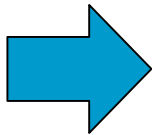
■ V850マイクロコントローラ(NECエレクトロニクス)

- μ T-Engine/V850E-MA3 (パーソナルメディア製)
 - 32ビットRISC CPU「V850E/MA3」 動作周波数 80MHz
 - CPU内蔵メモリ ROM 512KB / RAM 32KB
 - ボード搭載メモリ FLASH-ROM 1MB / SDRAM 8MB

対象CPUへ適応する場合の問題点

■ 問題点

- T-Kernel/SEの設計ポリシーによる問題点
 - メモリ管理はMMUによるページ管理が前提
- 対象CPUのコンパイラ(gcc)の制限
 - PIC対応されていない
 - 共有ライブラリが使用できない
 - プロセスロード方法の検討



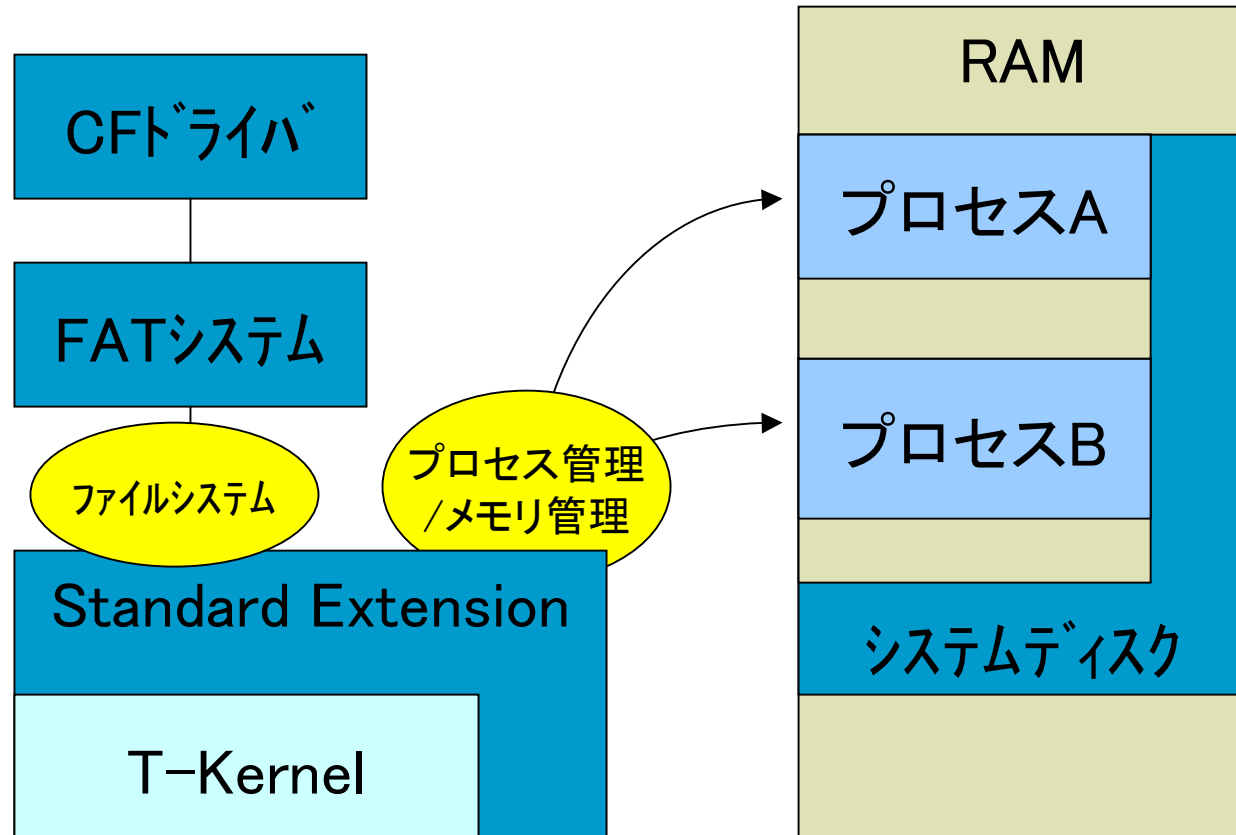
問題点を考慮し、MMU非搭載CPUへの
適応仕様を検討

MMU非搭載CPU向け適応仕様(1)

- プロセス管理(プロセスロード)
 - プロセスの配置アドレスは静的に固定
 - アドレス固定でのマルチプロセス対応
 - プロセスは実メモリイメージ
- メモリ管理
 - MMU依存部を削除し、物理メモリ空間に対応させる
- 共有ライブラリ
 - 適応しない
 - ※プロセスの動的ロードを考えていないため、使用不可でも問題ないとする。

MMU非搭載CPU向け適応仕様(2)

■ 想定する利用形態



適応化実装(1)

■ MMU依存部分の切り分け方

– #define VIRTUAL_ADDRESS

- プロセス管理、メモリ管理機能のMMU機能の取り外しに使用。一部MMULレス用処理を新規追加。

■ メモリ管理

– API(ライブラリコール)

- malloc/calloc/realloc/free(非常駐ローカルメモリ)
- Smalloc/Scalloc/Srealloc/Sfree(非常駐共有メモリ)

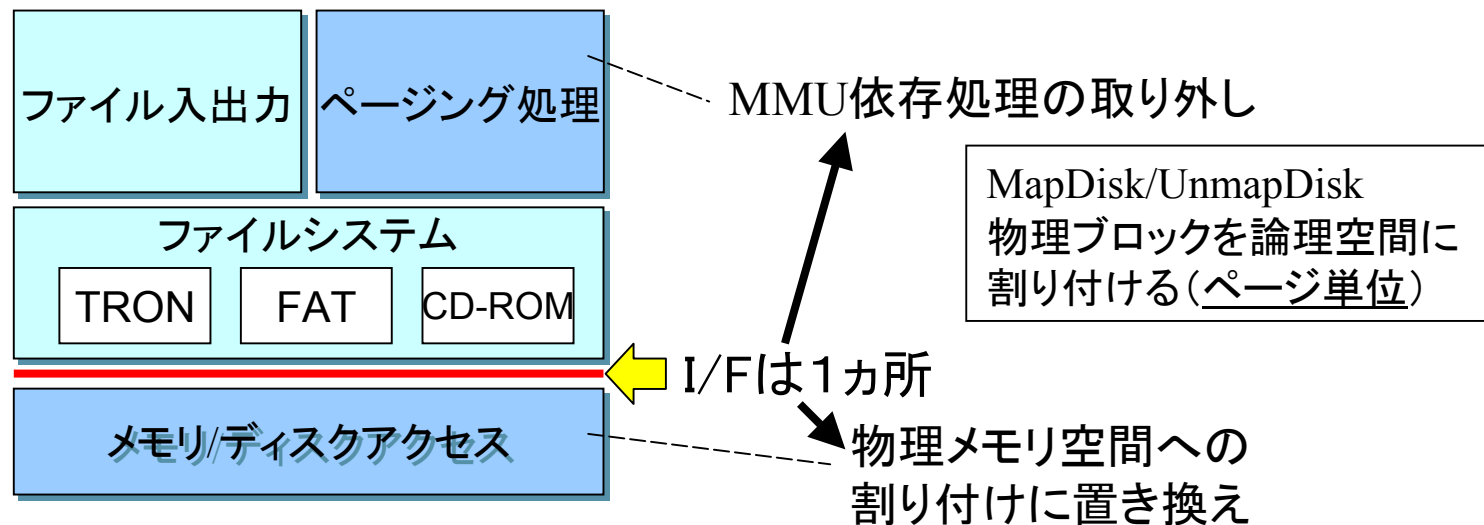
MMULレス版では、T-Kernel/OSの機能で代用

適応化実装(2)

■ ファイルシステム

– ページング(メモリ管理機能)と密接な構造

- 既存コードを極力生かす
 - ディスクをマップするメモリはページ単位のまま
- 新規コードは最小限
 - 物理ブロックを物理メモリにマップ/アンマップ



適応化実装(3)

■ 公開ドライバ

– システムディスクドライバ

- 動作確認用としてメモリディスクを使用した。
- 本質的には無改造で使用可能
 - 実際にはRAMディスク用の論理フォーマット処理を追加

– PCMCIAカードマネージャ

- 使用していないが、リンクエラー抑止のため、とりあえず他のCPUのものをリンク

– その他のドライバは未着手だが、MMUレスという観点での対処は不要。

- 「T-Engine デバイスドライバインタフェースライブラリ仕様」にしたがって実装すればよい

適応化結果

■ サイズ

FRの適応化結果

	OS適応時	SE適応時
text	77.1KB	401.6KB
data	1.5KB	5.5KB
bss	2.5KB	8.6KB

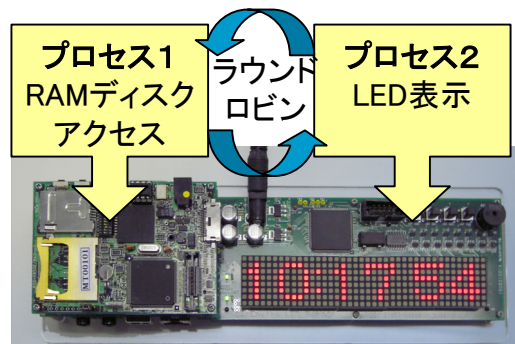
Standard ExtensionにはFAT分とユニコード/ترونコード変換テーブルを含む。

- MMULレスでは使用不可能な機能のコードも存在
→ まだまだ削減可能のはず

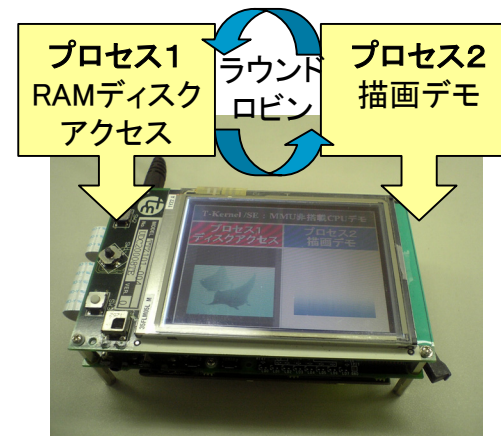
デモ内容

- V850、FRでStandard Extensionが動作しています。
 - ファイルアクセス(FAT)
 - マルチプロセス

V850(μ T-Engine)



FR(Ansel-Tea)



- トロン協会ブース(ブース番号1-1)にて展示中

まとめ

- MMU非搭載のシングルチップマイコンにおいてもT-Kernel/SEの機能が利用できる可能性が明らかになった。
 - ニーズの多いと思われるファイルシステムの動作実績が得られた。
 - 今後はTCP/IPへのニーズが出てくるものと考えられる。
 - フル規格のT-Engineとのスケーラビリティという点でのメリットも。

本活動のお問い合わせ先

- 本活動に関する、要望、コメントありましたら下記へ

社団法人トロン協会

E-mail: info@assoc.tron.org

〒108-0073 東京都港区三田1-3-39 勝田ビル5F

TEL: 03-3454-3191 FAX: 03-3454-3224