

TOPPERS Automotive Kernel  
アプリケーションノート  
m16c-renesas

Ver.2.00

2008/11/05



## TOPPERS Automotive Kernel

### Toyohashi Open Platform for Embedded Real-Time Systems Automotive Kernel

Copyright (C) 2006-2008 by Witz Corporation, JAPAN

上記著作権者は、以下の (1)~(4) の条件か、Free Software Foundation  
によって公表されている GNU General Public License の Version 2 に記  
述されている条件を満たす場合に限り、本ソフトウェア（本ソフトウェア  
を改変したものを含む、以下同じ）を使用・複製・改変・再配布（以下、  
利用と呼ぶ）することを無償で許諾する。

- (1) 本ソフトウェアをソースコードの形で利用する場合には、上記の著作  
権表示、この利用条件および下記の無保証規定が、そのままの形でソー  
スコード中に含まれていること。
- (2) 本ソフトウェアを、ライブラリ形式など、他のソフトウェア開発に使  
用できる形で再配布する場合には、再配布に伴うドキュメント（利用  
者マニュアルなど）に、上記の著作権表示、この利用条件および下記  
の無保証規定を掲載すること。
- (3) 本ソフトウェアを、機器に組み込むなど、他のソフトウェア開発に使  
用できない形で再配布する場合には、次のいずれかの条件を満たすこ  
と。
  - (a) 再配布に伴うドキュメント（利用者マニュアルなど）に、上記の著  
作権表示、この利用条件および下記の無保証規定を掲載すること。
  - (b) 再配布の形態を、別に定める方法によって、TOPPERS プロジェクトに  
報告すること。
- (4) 本ソフトウェアの利用により直接的または間接的に生じるいかなる損  
害からも、上記著作権者および TOPPERS プロジェクトを免責すること。

本ソフトウェアは、無保証で提供されているものである。上記著作権者お  
よび TOPPERS プロジェクトは、本ソフトウェアに関して、その適用可能性も  
含めて、いかなる保証も行わない。また、本ソフトウェアの利用により直  
接的または間接的に生じたいかなる損害に関しても、その責任を負わない。



< 目次 >

1. 概要.....	1
1.1. はじめに.....	1
1.2. 関連文書.....	1
2. 開発環境.....	2
2.1. サポートターゲット.....	2
2.2. 開発環境の構築.....	2
2.3. ビルド.....	2
2.4. FlashROMへの書込み.....	2
2.5. KD30 によるデバッグ.....	3
3. サンプルアプリケーション.....	4
3.1. ターゲットボードの設定.....	4
3.2. PCターミナルソフトの設定.....	4
3.3. オブジェクトの依存情報.....	4
4. カーネルのカスタマイズ.....	6
4.1. カテゴリ 1 割込み入り口生成抑止.....	6
4.2. カーネルスタックのサイズ.....	6
4.3. カーネルエラーステータス.....	7
変更履歴.....	8

## 1. 概要

### 1.1. はじめに

本アプリケーションノートは、ルネサステクノロジ社製開発環境を用いて、ルネサステクノロジ社製 M16C/20 シリーズマイコン上に構築された、TOPPERS Automotive Kernel サンプルアプリケーションの利用方法およびカスタマイズ項目を記載します。なお、本アプリケーションノートで記載するのは、カーネル機能の使用法の一例であり、安全なシステム構築を保障するものではないことをご了承ください。

### 1.2. 関連文書

カーネルの機能詳細は「TOPPERS Automotive Kernel 外部仕様書」を参照してください。

カーネル用 SG の使用方法是「TOPPERS Automotive Kernel SG 取扱説明書」を参照してください。

機種に依存しないアプリケーション情報は「TOPPERS Automotive Kernel アプリケーションノート」を参照してください。

OSEK/VDX 仕様：OSEK/VDX Operating System Ver2.2.1

OSEK/VDX OIL Specification Version 2.5

OSEK/VDX Binding Specification Version 1.4.2

※OSEK/VDX が公開している各仕様書は <http://www.osek-vdx.org/> よりダウンロード可能です。

## 2. 開発環境

TOPPERS Automotive Kernel およびサンプルアプリケーションを開発するための、開発環境の構築手順について説明します。

### 2.1. サポートターゲット

以下にサポートしているターゲットの一覧を記載します。

メーカー名	ターゲットシステム名
オックス電子	oaks16-mini Fullkit

### 2.2. 開発環境の構築

開発環境には「oaks16-mini Fullkit」に同封されている統合開発環境 (TM) およびコンパイラパッケージ (NC30WA) を使用します。

同封されている CD より開発環境をインストールします。CD が自動起動すると WEB ページ (CD ローカル) が表示されますので、WEB ページより TOOLS(ディレクトリ)を選択します。TOOLS をクリックすると CD 内の TOOL 階層が表示されます。「TM」⇒「setup」と階層をたどり、統合開発環境をインストールします。続いて、「NC30WA」⇒「setup」と階層をたどり、コンパイラパッケージをインストールします。最後に「Flashstart」階層をそのままご使用のパソコンの、任意の階層にコピーします。

これで開発環境のインストールは終わりですが、必要に応じてシリアルデバッガ (KD30) をインストールしておくと、デバッグ時には便利です。

### 2.3. ビルド

統合開発環境を起動し、「¥¥tools¥m16c-renesas¥oaks16-mini」にあるプロジェクトファイル (.tmk ファイル) を開きます。ProjectEditor を開き、「プロジェクト」⇒「依存関係の更新」メニューにて依存関係を更新します。TM ツールバーのビルドボタンを押下し、ビルドを実行します。

これでサンプルプログラムのビルドは終わりです。

なお、「output」階層にロードモジュールファイル (.mot ファイル) が生成されない場合は、ProjectEditor の「プロジェクト」⇒「情報」メニューよりプロジェクトプロパティを開き、ツールタブで表示されるページの中央にある「lmc30」にチェックが付いているか確認してください。「lmc30」はロードモジュールコンバータツールです。

### 2.4. FlashROMへの書き込み

開発用のパソコンと OAKS16-MINI ボードとを、RS-232C ケーブルを使って接続します。CNVSS ピンを、ジャンパーピンを使ってショートさせ、電源を投入します。

FlashSta を起動し、「Select Program」メニューにて使用するポートを選択して「OK」ボタンを押下します。その後、「ID Check」メニューが表示されたら「Refer..」ボタンを押下し、生成した MOT

ファイルを選択します。「Select Program」メニューに戻ったら、「OK」ボタンを押下します。その後 M16C Flash Start メニューを表示されます。

まず、「Erase」ボタンを押下し、フラッシュ ROM をクリアします。その後、「Program..」ボタンを押下しプログラムの書込みを行います。書込みが終了したら、「Exit」ボタンを押下し FlashSta を終了させます。電源を切り、CNVSS ピンからジャンパーピンを外します。

## 2.5. KD30 によるデバッグ

モニターデバッガ (KD30) にてデバッグを行うことが可能です。なお、KD30 の使用方法についてはマニュアルを参照してください。

KD30 はシリアル通信を用いたデバッガです。このため、サンプルプログラムで使用しているシリアル通信機能を使用すると、デバッガの動作を阻害してしまいます。KD30 を使用する場合、開発環境にて「USE\_KD30」コンパイルオプションを定義し、再度ビルドを実施してください。

### 3. サンプルアプリケーション

サンプルアプリケーションを動作させるためのターゲットボードや PC 上のターミナルソフトの設定、オブジェクトの依存情報について説明します。

#### 3.1. ターゲットボードの設定

サンプルアプリケーションにおけるコマンドの入力およびログの出力は、UART（シリアル I/O）のチャンネル 1 を使用しています。オックス電子社製 OAKS16-MINI FULLKIT ボード上の D-Sub9Pin コネクタから PC とターゲットボードをストレートケーブルにて接続することで通信が可能となります。

サンプルアプリケーションにおけるアプリケーションモード入力用のデバイスとして、SW4(ポート 8 の Bit1)および SW5(ポート 8 の Bit0)を使用しています。各 SW の入力状態により電源投入時に異なるアプリケーションモードにてカーネルが起動します。各 SW の入力状態とアプリケーションモードの関係を以下に記載します。

SW4	SW5	アプリケーションモード
OFF	OFF	アプリケーションモード 1
OFF	ON	アプリケーションモード 2
ON	OFF	アプリケーションモード 3
ON	ON	アプリケーションモード 3

サンプルアプリケーションにおけるシステムタイマはタイマ A0 で実装しています。

サンプルアプリケーションにおけるカウンタソース割込みはタイマ A1 で実装しています。

サンプルアプリケーションにおける HW カウンタ用割込みはタイマ B0 およびタイマ B1 で実装しています。

#### 3.2. PCターミナルソフトの設定

サンプルアプリケーションを動作させるための設定を以下に記載します。

設定項目	設定内容
通信ボーレート	38400bps
データサイズ	8bit
パリティチェック	なし
ストップビット	1bit
フロー制御	なし

#### 3.3. オブジェクトの依存情報

サンプルアプリケーションで定義しているオブジェクトの内、ターゲットシステムに依存する項目の一覧を以下に記載します。

マイコン M16C/26 では内臓 RAM は 2KByte となっており、カーネルおよびサンプルアプリケーション

ン全機能を入れると RAM 領域をオーバーしてしまいます。その中でもエラーフックルーチンでは多くの RAM を消費しているため、oaks16-mini ではエラーフックルーチンを無効に定義しています。

オブジェクト種別	オブジェクト概要	オブジェクトパラメータ
OS	OS システム	スタートアップフック：使用 シャットダウンフック：使用 エラーフック：未使用 プレタスクフック：未使用 ポストタスクフック：未使用 スケジューラリソース：使用
タスク	メインタスク	スタックサイズ：0x0100
	最高優先度タスク	スタックサイズ：0x00A0
	ノンプリエンプティ ブタスク	スタックサイズ：0x00A0
	同時実行タスク 1	スタックサイズ：0x00A0
	同時実行タスク 2	スタックサイズ：0x00A0
	同時実行タスク 3	スタックサイズ：0x00A0
	同時実行タスク 4	スタックサイズ：0x00A0
	同時実行タスク 5	スタックサイズ：0x00A0
ISR	シリアル IO 受信割 込み	優先度：6 ベクタエントリ番号：20
	ハードウェアカウン タ 1 割込み	優先度：5 ベクタエントリ番号：26
	システムタイマ割込 み	優先度：4 ベクタエントリ番号：21
	カウンタソース割込 み	優先度：3 ベクタエントリ番号：22
	ハードウェアカウン タ 2 割込み	優先度：2 ベクタエントリ番号：27



## 4. カーネルのカスタマイズ

TOPPERS Automotive Kernel は、ターゲットシステム oaks16-mini Fullkit 上で動作するように設定されています。しかし、ハードウェアの改良やシステム規模の変更に対し、カーネルの設定を必要とする場合があります。このため、ユーザーにてカスタマイズ可能な項目について以下に記載します。

### 4.1. カテゴリ 1 割込み入り口生成抑止

OSEK 仕様において ISR カテゴリ 1 は、OS が介在せずより高速に処理される割込みとして定義されています。しかし、カテゴリ 1 の割込みを使用するためには、開発環境（コンパイラ）に対して割込み関数指定するオプションなどターゲット固有の知識を必要とします。このため、TOPPERS Automotive Kernel ではユーザーが普通の関数と同じように割込み関数を作成できるよう、割込み出入り口処理とそこからユーザー関数を呼出す処理を生成するようにしています。

しかし、これは OSEK 仕様でいわれている高速に処理される割込みという意図を阻害するものになります。このため、ターゲット環境を理解し、高速に処理される割込みを生成する場合のために、「OMIT\_ISR1\_ENTRY」コンパイルオプションを有効とすることで、上記割込み出入り口処理の生成を抑止します。なお、本機能を使用した場合、ユーザー割込み関数を「#pragma INTERRUPT」機能にて割込み関数指定する必要があります。

本コンパイルオプションはカーネルコードの「cpu\_defs.h」内に定義されています。

また、カーネル機能を使用するためのヘッダファイル「kernel.h」をインクルードすることで、上記「cpu\_defs.h」も内部からインクルードされます。このため、上記コンパイルオプションの有効・無効はユーザーコード内で参照可能です。

### 4.2. カーネルスタックのサイズ

TOPPERS Automotive Kernel ではシステムが十分に動作できるよう、カーネルスタックは余裕を持ったサイズを設定しています。しかし、ターゲットアプリケーションによっては、それでも不足している場合、逆に余剰であり他の RAM 領域として使用したい場合があります。

カーネルスタックのサイズおよびスタックセクションの先頭アドレスは、セクション定義ファイル「sectm16c.inc」にて設定しています。また、マイコン M16C/26 では、0x0400～0x0BFF に内蔵 RAM がマップされており、0x0BFF をスタックセクション末尾として必要サイズ分領域を確保しています。

カーネルリリース時は「sectm16c.inc」ファイル内の 63 行目「.blkb」の行にカーネルスタックサイズとして 100H(256Byte)と定義され、62 行目「.org」の行にカーネルスタック領域の先頭アドレスを B00H と定義しています。

このため、カーネルスタックのサイズを修正する場合は、上記箇所にそれぞれ必要な値を設定してください。なお、「スタックセクションの先頭アドレス + スタックサイズ」の値が、スタックセクションの次のアドレスである C00H になるようにしてください。

### 4.3. カーネルエラーステータス

TOPPERS Automotive Kernel ではシステムコール実行時のエラーチェックは、拡張エラーまでチェックを実施する仕様となっています。OSEK 仕様では OIL にてチェックするエラーステータスを選択（標準・拡張）できるようになっていますが、SG 実行時のリコンパイル対象を少なくするために、SG としては未対応としています。

なお、拡張エラーチェックを取り外したい場合のために、「BASIC\_STATUS」コンパイルオプションを有効とすることで、標準エラーのチェックのみを実施するように変更が可能です。

本コンパイルオプションはカーネルコードの「sys\_defs.h」内に定義されています。

文書番号：

仕様書名：TOPPERS Automotive Kernel アプリケーションノート m16c-renesas Ver.2.00

2008/11/05



#### 変更履歴

Version	Date	Detail	Editor
1.00	2006/05/30	・ 新規作成	ヴィッツ
2.00	2008/11/05	・ カーネル名称変更 ・ 部署名変更	ヴィッツ