

OAKS16-62P
TOPPERS/ASP
ユーザーズ・マニュアル

安全設計に関するお願い

・弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

- ・本資料は、お客様が用途に応じた適切な製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてオックス電子および情報を提供いただいた各社が所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
- ・本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、オックス電子は責任を負いません。
- ・本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、オックス電子は特性改良などにより予告なしに変更することがあります。
- ・本資料に記載の図、表に示す技術的な内容、及びプログラム、アルゴリズムを流用する場合、お客様の責任において実施してください。また、組み込んだプログラム、アルゴリズム単体で評価するだけでなく、システム全体で十分に評価してください。オックス電子は、一切責任を負いません。
- ・本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、オックス電子へご照会ください。
- ・本資料の転載、複製については、文書によるオックス電子の事前の承諾が必要です。
- ・本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらオックス電子までご照会ください。

目次

1.はじめに	4
1.1.必要な環境	4
1.2. CD の内容	5
2.開発環境のセットアップ	6
2.1.その他注意事項	6
3.TOPPERS について	7
3.1.TOPPERS プロジェクトとは	7
3.2.TOPPERS/ASP カーネルについて	7
3.3.更に詳しく学ぶには	7
4.TOPPERS/ASP カーネルのビルド方法	8
4.1.サンプルプログラムについて	8
4.2.TOPPERS/ASP カーネルのビルド	9
4.2.1.ワークスペースのプロジェクトについて	10
4.2.2.ビルド設定について	11
4.2.3.makeoffset プロジェクトの作業	12
4.2.4.kernel プロジェクトの作業	13
4.2.5.cfg プロジェクトの作業	15
4.2.6.sample1 プロジェクトの作業	17
4.3.デバッグの設定	20
4.4.ターゲットボードの設定	21
4.5.デバッグの開始	22
4.6.ハイパーターミナルの設定	24
4.7.プログラムの実行	26
4.8.サンプルプログラムの動作	27
5.TOPPERS/ASP カーネル技術資料	29
5.1.フォルダ構成	29
5.2.ドキュメントについて	29
5.3.参考資料	29
6.補足資料	30
6.1.ソースコードの修正について	30
6.2.メモリマップの状況	31

1.はじめに

このマニュアルは、OAKS16-62P boardkit と OAKS Multi I/O Board を組み合わせて、TOPPERS/ASP カーネルを動作させるためのソフトウェアマニュアルです。

TOPPERS プロジェクトの TOPPERS/ASP カーネル **RSK-M16C-62P** (ルネサステクノロジー) 簡易パッケージ [asp_rsk_m16c_62p_hew-20091201.lzh](#) をベースに、ルネサステクノロジー製 E8a エミュレータ環境にて実行できるようにしたものです。

このマニュアルに沿って、学習することで、TOPPERS/ASP カーネルと初めて TOPPERS に触れる方が動作させるサンプルプログラムをコンパイル・実行させることができるようになります。

1.1.必要な環境

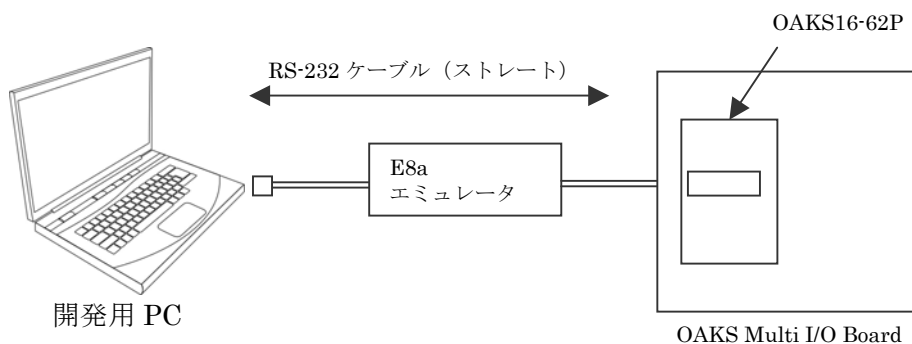
ここでは、下記の環境で、TOPPERS/ASP カーネルを動作させることを想定しています。

ターゲットボード:

OAKS16-62P boardkit
OAKS Multi I/O Board
ルネサステクノロジー製 E8a エミュレータ

開発用 PC

ルネサステクノロジーの High-performance Embedded Workshop が動作する PC Sample プログラムの動作にシリアルポートが 1 つ必要となります。
また接続のために RS-232C ケーブル (ストレート) が必要です。



1.2. CD の内容

CDROM には TOPPERS/ASP カーネルのプログラム開発に必要なソースコード、電子マニュアルなどが含まれています。

以下に CDROM の構成を示します。

CDROM	----	oaks16-62p	----	asp	<TOPPERS/ASP ソースコード一式>
			----	doc	<ドキュメント>

2.開発環境のセットアップ

ここでは、すでに開発環境がインストールされているものとして説明を行います。バージョンが古い場合には、オートアップデートにて更新してください。

また、インストール作業に関しては、ユーザ責任で行ってください。

このマニュアルを作成した時点でのバージョンは次のとおりです。

- HEW (4.07.00.007)
- M16C Series, R8C Family C Compiler (5.45.00)
- M16C E8a Emulator Debugger (1.03.02)

開発環境のインストールやアップデートに関する説明は OAKS16-62P ユーザーズマニュアルを参照してください。

2.1.その他注意事項

この TOPPERS/ASP カーネルを動作させるには、あらかじめ、OAKS16-62P のプログラミングに関する知識が最低限必要とされます。

初めて、この文書を読まれる方は、まずプログラミング・マニュアルを参照して、サンプルプログラムなどを実行してみることをお勧めいたします。

[自習用テキスト](#)をご用意していますので、参考にしてください。

本マニュアルでは、TOPPERS/ASP について、すべての説明を行うことはできませんので、ご了承願います。

TOPPERS/ASP カーネルや sample1 プログラムに関しては、TOPPERS プロジェクトのドキュメントを参照するなどしてください。オークス電子では、サポートいたしません。

3.TOPPERS について

3.1.TOPPERS プロジェクトとは

TOPPERS (Toyohashi OPen Platform for Embedded Real-time Systems) プロジェクトは、ITRON 仕様の技術開発成果を出発点として、組み込みシステム構築の基盤となる各種のソフトウェアを開発し、良質なオープンソースソフトウェアとして公開することで、組み込みシステム技術と産業の振興を図ることを目的としたプロジェクトです。また、教育コースや教材の開発と、それを用いた教育の場を提供するなどの活動を通じて、組み込みシステム技術者の育成に貢献することも目的としています。

TOPPERS プロジェクトは、2003 年 9 月に設立した特定非営利活動法人(NPO 法人)を中心に、名古屋大学教授の高田広章氏をリーダーとして、産学官の団体と個人の連携により推進しています。

3.2.TOPPERS/ASP カーネルについて

TOPPERS/ASP カーネルは、TOPPERS 新世代カーネルの基盤(出発点)となるリアルタイムカーネルで、TOPPERS 新世代カーネル統合仕様に準拠しています。ASP (Advanced Standard Profile) の名前が示す通り、 μ ITRON4.0 仕様のスタンダードプロファイル準拠のリアルタイムカーネルである TOPPERS/JSP カーネルを拡張・改良する形で開発しました。

3.3.更に詳しく学ぶには

TOPPERSプロジェクトの[ホームページ](#)を参照してください。

具体的なプログラミングやTOPPERS/ASPについての詳細は、[TOPPERSプロジェクト](#)の教育用コンテンツなどで学習を行うことができます。

さらに TOPPERS/ASP カーネルを理解するために、 μ ITRON4.0 の仕様書が参考になることでしょう。

また、CDROM に μ ITRON4.0 仕様書も添付してありますので参考にしてください。

このマニュアルと提供している CDROM のソースコードは、TOPPERS プロジェクトの成果物をベースに、OAKS16-62P で動作するように修正したものです。

ソースコードの修正内容については、本マニュアルの最期で説明します補足資料の内容が参考になることでしょう。

4.TOPPERS/ASP カーネルのビルド方法

この章では、TOPPERS/ASP カーネルのソースコード、そしてサンプルプログラムのビルド方法について説明いたします。

4.1. サンプルプログラムについて

サンプルプログラムは、TOPPERS/ASP カーネルの基本的な動作を確認するために公式に添付されているものです。

サンプルプログラムを実行させると、ターゲットボード OAKS16-62P よりシリアル I/O 出力されます。3つのタスクが実行され、メインタスクは、シリアル I/O ポートからの文字入力を行います。

そして、入力された文字に対応した処理を実行します。例えばレディーキューを回転する'r'を入力すると、task1 から task2 へと実行状態のタスクが切り替わります。

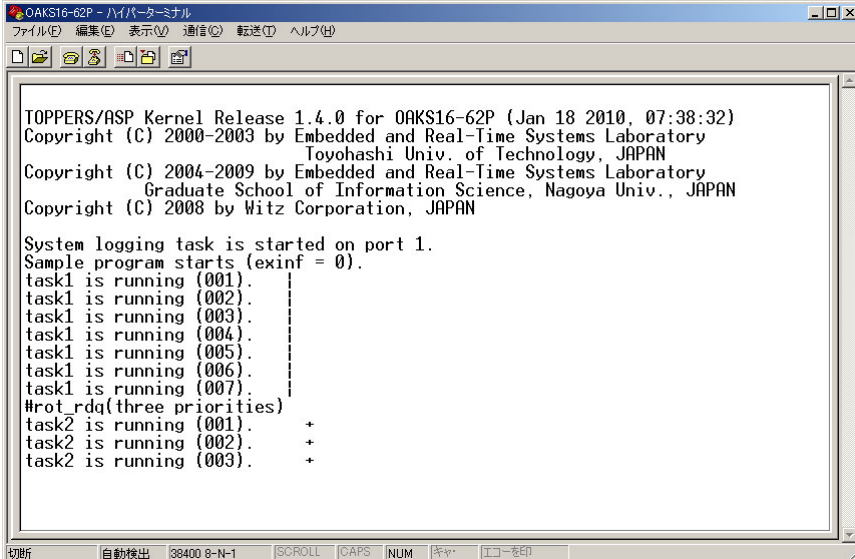
他にも次のようなものがあります。

- '1': 対象タスクを TASK1 に切り換える(初期設定)。
- '2': 対象タスクを TASK2 に切り換える。
- '3': 対象タスクを TASK3 に切り換える。

Control-C または'Q'が入力されると、プログラムを終了します。

起動中のタスクに関する指示など詳細については、ソースコードの sample1.c のコメントを参照してください。

ここでは、まずサンプルプログラムを実行させることに専念してみましょう。



```
OAKS16-62P - ハイパーターミナル
ファイル(F) 編集(E) 表示(V) 通信(C) 転送(T) ヘルプ(H)
TOPPERS/ASP Kernel Release 1.4.0 for OAKS16-62P (Jan 18 2010, 07:38:32)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
Toyohashi Univ. of Technology, JAPAN
Copyright (C) 2004-2009 by Embedded and Real-Time Systems Laboratory
Graduate School of Information Science, Nagoya Univ., JAPAN
Copyright (C) 2008 by Witz Corporation, JAPAN

System logging task is started on port 1.
Sample program starts (exinf = 0).
task1 is running (001). |
task1 is running (002). |
task1 is running (003). |
task1 is running (004). |
task1 is running (005). |
task1 is running (006). |
task1 is running (007). |
#rot_rdq(three priorities)
task2 is running (001). +
task2 is running (002). +
task2 is running (003). +

切替所 自動検出 38400 8-N-1 SCROLL CAPS NUM キー エコーを印
```

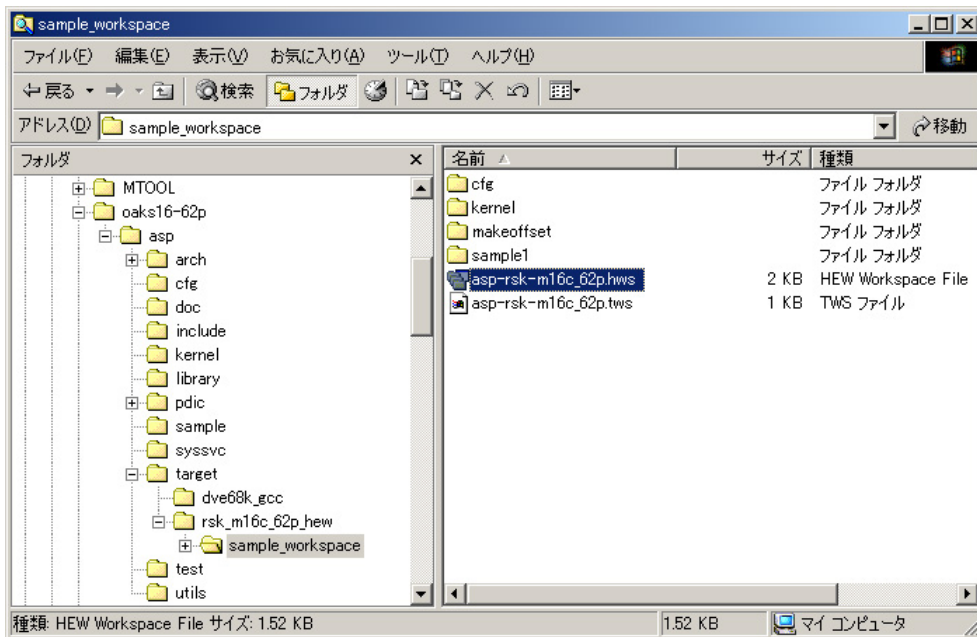
4.2.TOPPERS/ASP カーネルのビルド

それでは、TOPPERS/ASP カーネルのビルドを行います。

まず、TOPPERS/ASP カーネルを収められている CD から、フォルダ『oaks16-62p』一式を開発用 PC の Cドライブにコピーします。

サンプルプログラムおよびカーネル・ビルドのためのサンプルワークスペースを開きます。

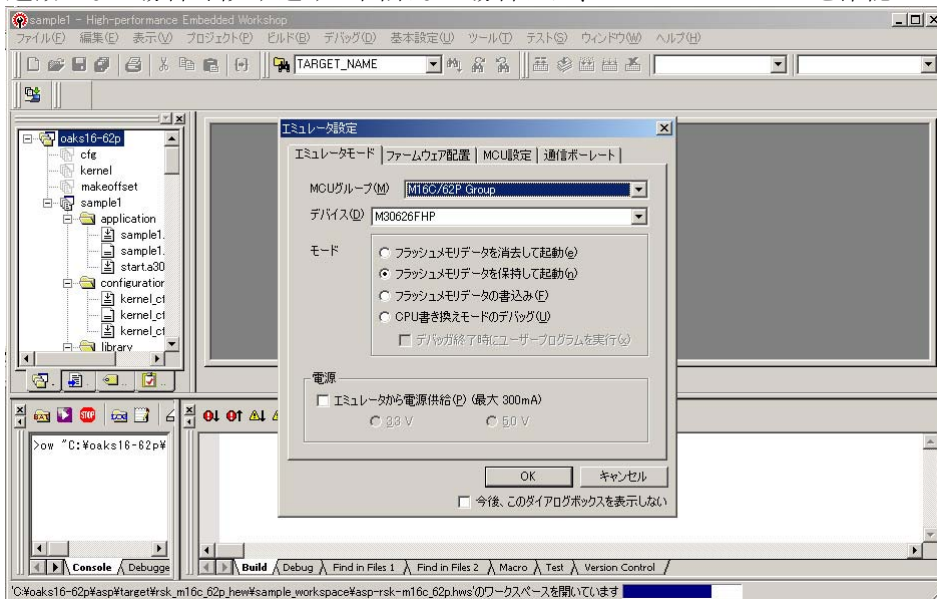
c:\oaks16-62p\asp\target\rsk_m16c_62p_hew\sample_workspace
のフォルダにあります。



asp-rsk-m16c_62p.hws というワークスペースファイルをダブルクリックします。

すると、次のような HEW の画面になります。

起動しない場合や読み込みが出来ない場合には、HEW のバージョンを確認してください。



エミュレータ設定のウィンドウが表示されますが、今は『キャンセル』をクリックしてください。

4.2.1.ワークスペースのプロジェクトについて

先ほどまでの手順で、TOPPERS/ASP カーネルのサンプルプロジェクトを開くことができました。

ワークスペースには、4つのプロジェクトがあります。

- `makeoffset`
- `kernel`
- `cfg`
- `sample1`

カーネル本体は、`kernel` プロジェクトとなります。

サンプルプログラム本体は、`sample1` プロジェクトです。

`makeoffset` と `cfg` プロジェクトは、TOPPERS/ASP カーネルを構成するために必要なプロジェクトです。ここでは詳細について触れません。

ただ、3つのプロジェクトによって、カーネルのビルドができるということと、通常、ユーザーがプログラムを作成する場合には、`sample1` プロジェクトのように別プロジェクトとして作業をするということを理解していただければよいでしょう。

カーネルのビルドの手順は次のようになります。

- ① `makeoffset` のビルド
- ② `kernel` のビルド
- ③ `cfg` のビルド

各プロジェクト事にビルドをするので、アクティブプロジェクトを切り替えて作業することになります。

4.2.2.ビルド設定について

プロジェクトのビルドの設定は、目的によって、`Debug` と `Release` が選択できます。

TOPPERS/ASP カーネルのソースコードでは、ビルド設定に `Debug` を選択すると、`USE_KD3083` が定義されているため、シリアル I/O 出力は行われなくなります。(KD3083 でデバッグをするために、シリアルポートを占有するため)

今回、デバッグ環境として、E8a エミュレータを使用しているということと、`sample1` プログラムは、シリアル I/O 入出力を使用するため、すべて `Release` を選択します。

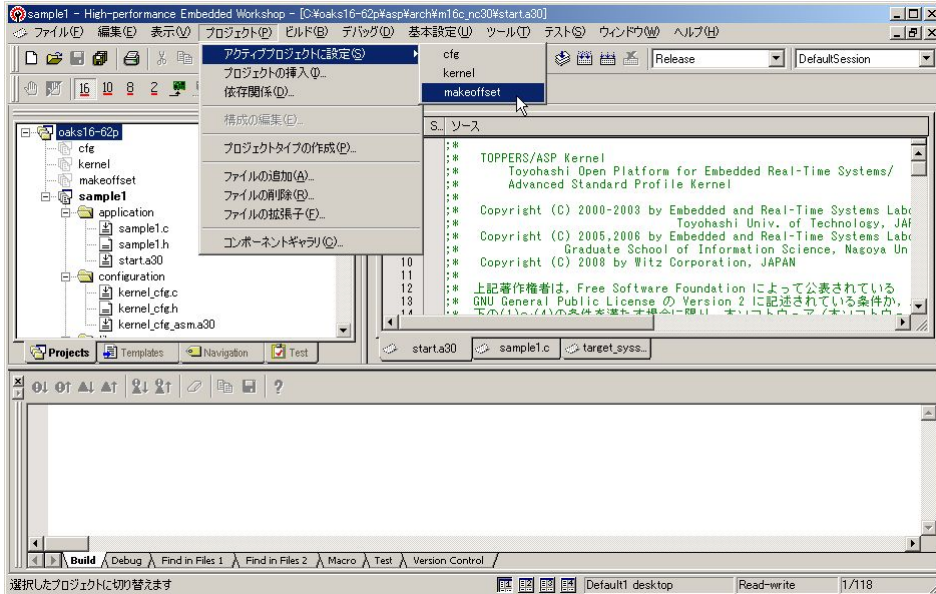
サンプルプロジェクトのビルド設定をデフォルト `Release` に設定しているため、通常は変更する必要はありません。

4.2.3.makeoffset プロジェクトの作業

それでは、makeoffset プロジェクトのビルドを行います。

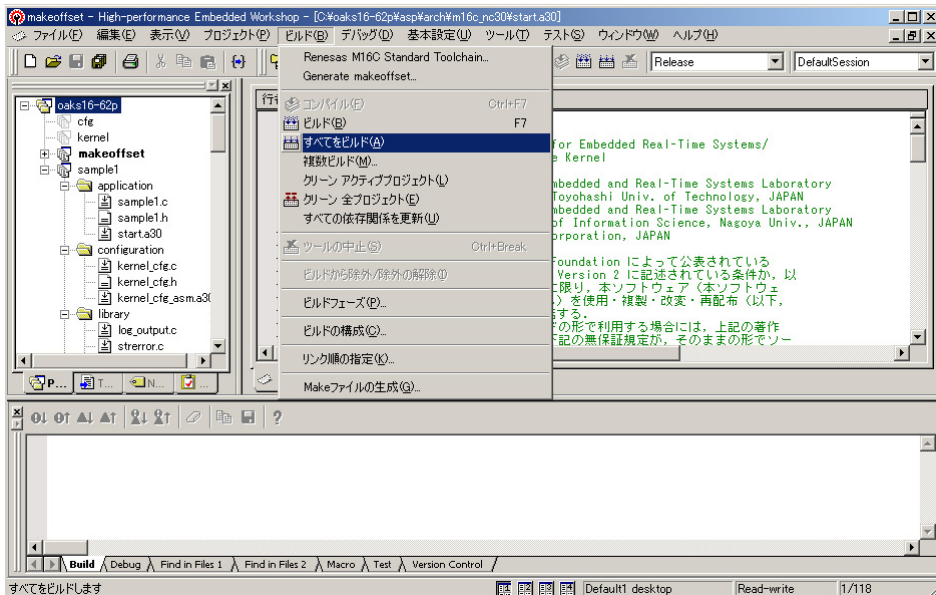
まず、ビルドを行うために、アクティブプロジェクトに設定します。そして、ビルドという手順になります。手順は次のとおりです。

①makeoffset プロジェクトをアクティブプロジェクトに設定します。



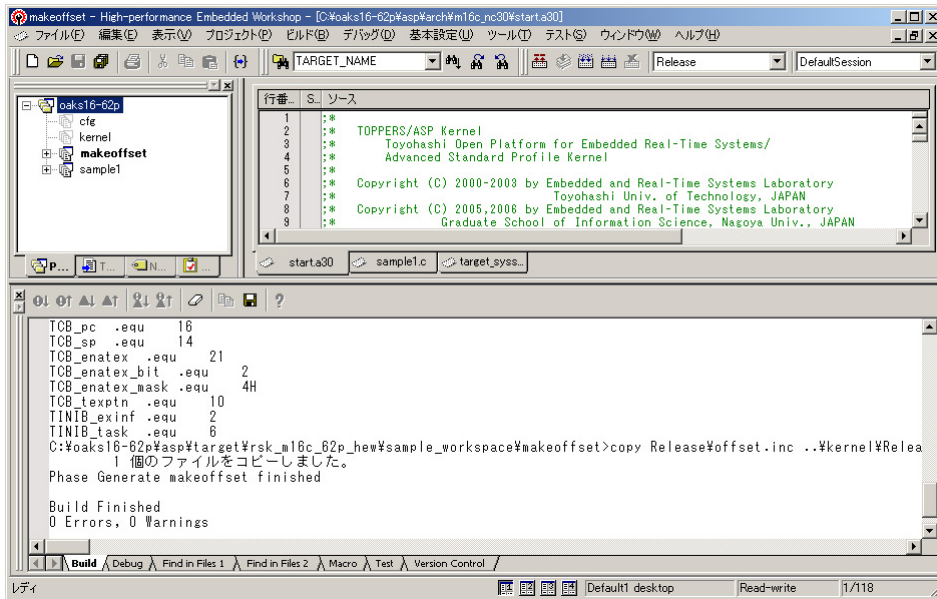
メニュー→プロジェクト→『アクティブプロジェクトに設定』から『makeoffset』を選択します。

②プロジェクトのビルド



メニュー→ビルド→『すべてをビルド』をクリックします。

しばらくすると、ビルドが完了して、下記のような画面になります。



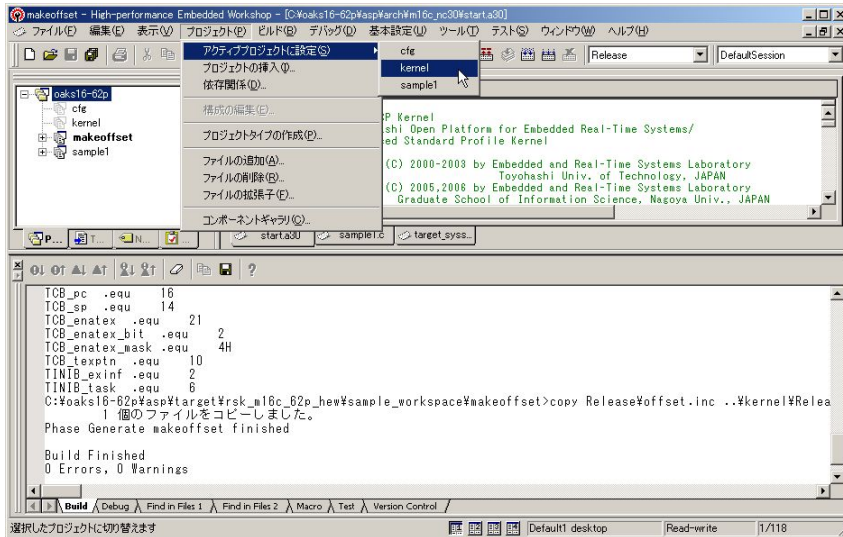
```
makeoffset - High-performance Embedded Workshop - [C:\oaks16-62p\asp\arch\m16c_nc30\starta30]
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) 基本設定(O) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)
TARGET_NAME Release DefaultSession
oaks16-62p
  cte
  kernel
  makeoffset
  sample1
行番 S. ソース
1  :*
2  :*
3  :*  TOPPERS/ASP Kernel
4  :*  Toyohashi Open Platform for Embedded Real-Time Systems/
5  :*  Advanced Standard Profile Kernel
6  :*
7  :*  Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
8  :*  Toyohashi Univ. of Technology, JAPAN
9  :*  Copyright (C) 2005,2006 by Embedded and Real-Time Systems Laboratory
   :*  Graduate School of Information Science, Nagoya Univ., JAPAN
starta30 sample1.c target_sys...
TCB_pc .equ 16
TCB_sp .equ 14
TCB_enatex .equ 21
TCB_enatex_bit .equ 2
TCB_enatex_mask .equ 4H
TCB_texptn .equ 10
TINIB_exinf .equ 2
TINIB_task .equ 6
C:\oaks16-62p\asp\target\rsk_m16c_62p_hew\sample_workspace\makeoffset>copy Release\offset.inc ..\kernel\Relea
1 個のファイルをコピーしました。
Phase Generate makeoffset finished
Build Finished
0 Errors, 0 Warnings
Build Debug Find in Files 1 Find in Files 2 Macro Test Version Control /
レディ Default1 desktop Read-write 1/118
```

makeoffset プロジェクト、ビルドが完了した様子。

4.2.4.kernel プロジェクトの作業

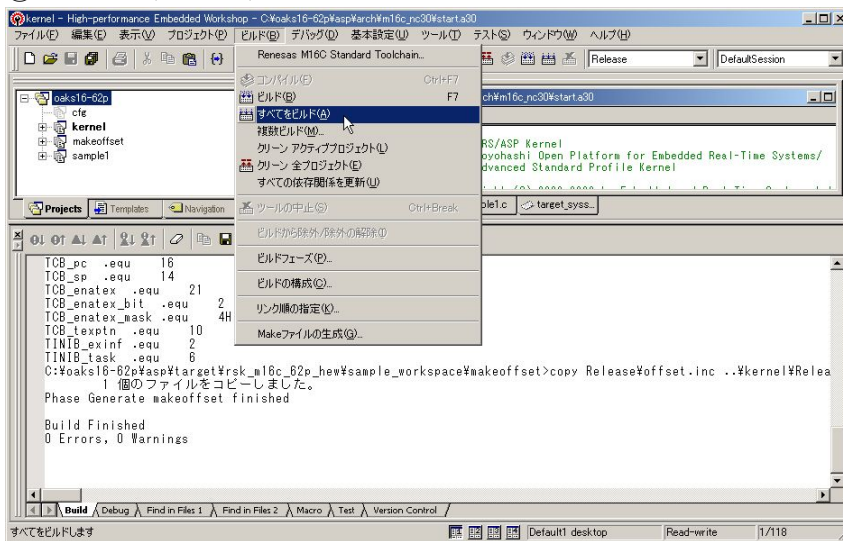
それでは、kernel プロジェクトのビルドを行います。
まず、ビルドを行うために、アクティブプロジェクトに設定します。そして、ビルドという手順になります。
手順は次のとおりです。手順は、makeoffset プロジェクトのときと同様です。

①kernel プロジェクトをアクティブプロジェクトに設定します。



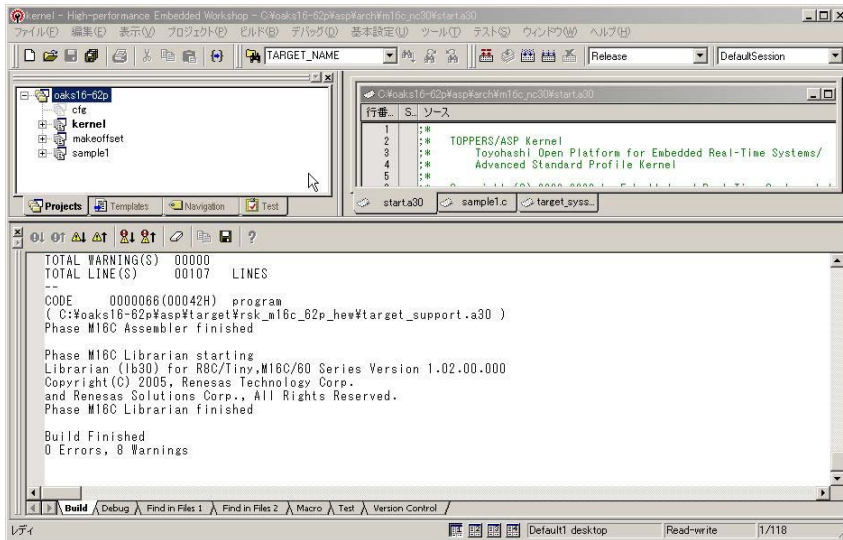
メニュー→プロジェクト→『アクティブプロジェクトに設定』から『kernel』を選択します。

②プロジェクトのビルド



メニュー→ビルド→『すべてをビルド』をクリックします。

しばらくすると、ビルドが完了して、下記のような画面になります。

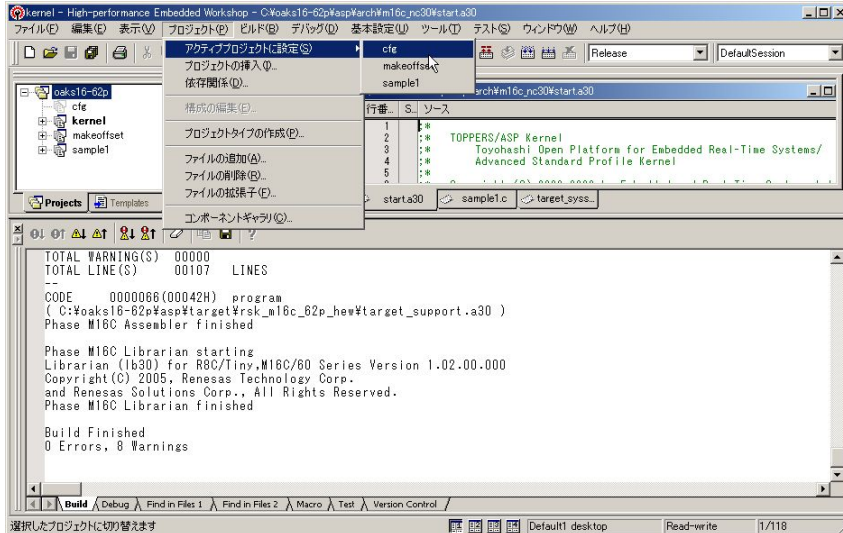


kernel プロジェクトのビルドが完了した様子。

4.2.5.cfg プロジェクトの作業

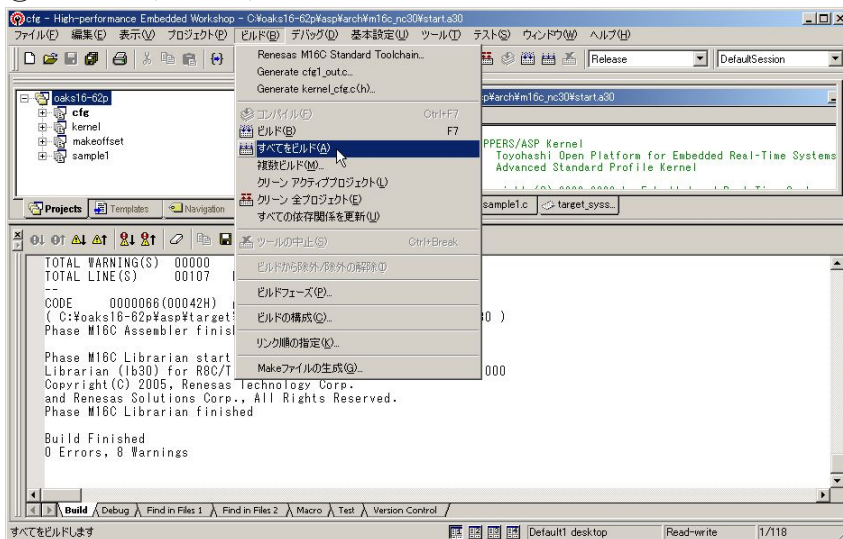
それでは、**cfg** プロジェクトのビルドを行います。
まず、ビルドを行うために、アクティブプロジェクトに設定します。そして、ビルドという手順になります。
手順は次のとおりです。手順は、**kernel** プロジェクトのときと同様です。

①**cfg** プロジェクトをアクティブプロジェクトに設定します。



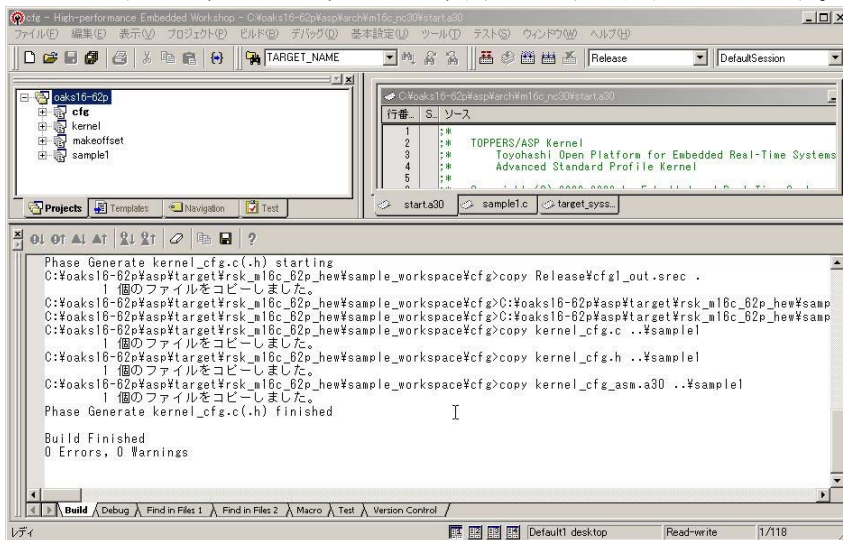
メニュー→プロジェクト→『アクティブプロジェクトに設定』から『**cfg**』を選択します。

②プロジェクトのビルド



メニュー→ビルド→『すべてをビルド』をクリックします。

しばらくすると、ビルドが完了して、下記のような画面になります。



cfg プロジェクトのビルドが完了した様子。

ここまでの作業で、TOPPERS/ASP カーネル本体のビルドは終了です。

4.2.6.sample1 プロジェクトの作業

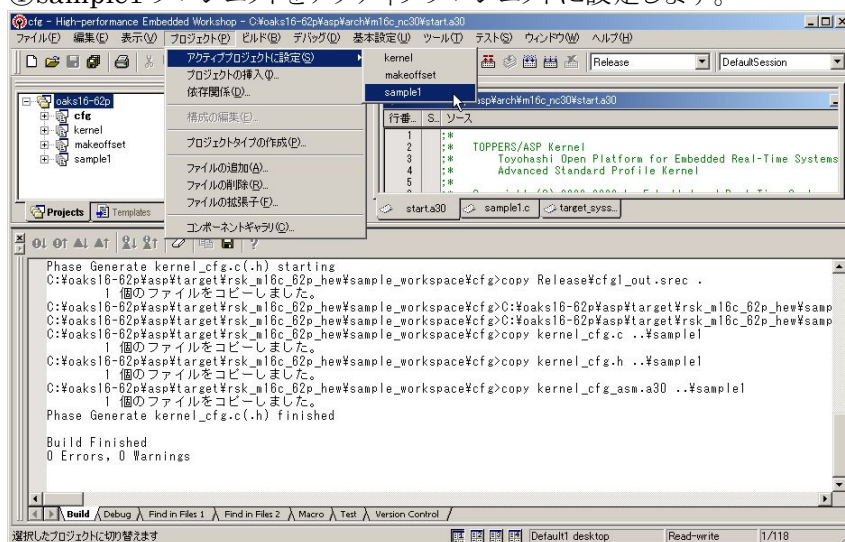
この sample1 プロジェクトは、サンプルプログラム本体のソースコード sample1.c があります。

それでは、最後に sample1 プロジェクトのビルドを行います。

まず、ビルドを行うために、アクティブプロジェクトに設定します。そしてエミュレータの設定をした後にビルドを行います。

手順は次のとおりです。

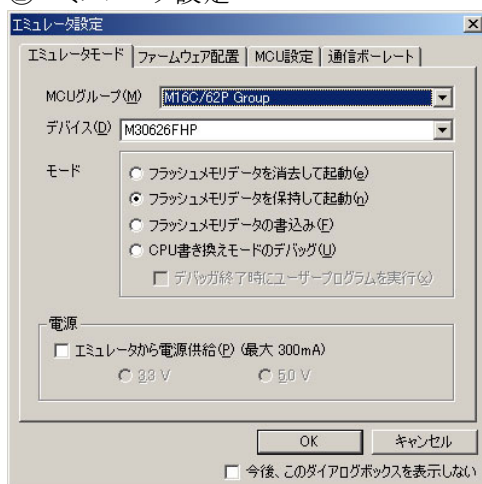
①sample1 プロジェクトをアクティブプロジェクトに設定します。



メニュー→プロジェクト→『アクティブプロジェクトに設定』から『sample1』を選択します。

エミュレータの設定画面が表示されます。

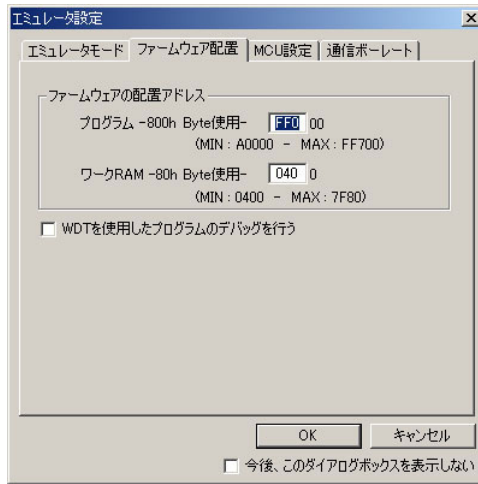
②エミュレータ設定



MCU グループ、デバイスを設定します。
MCU グループ: M16C/62P Group
デバイス: M30626FHP

エミュレータモードを設定します。

③ファームウェア配置の設定

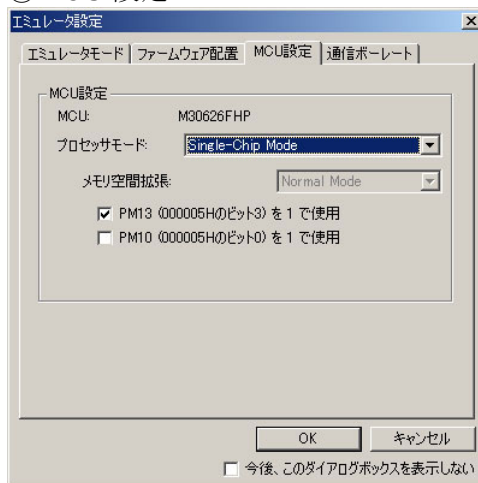


ファームウェアの配置アドレス
プログラム: FF00
ワーク RAM: 0400

どこにエミュレータのファームウェアを配置するかを設定します。

実行ファイルとぶつからないエリアを使用します。

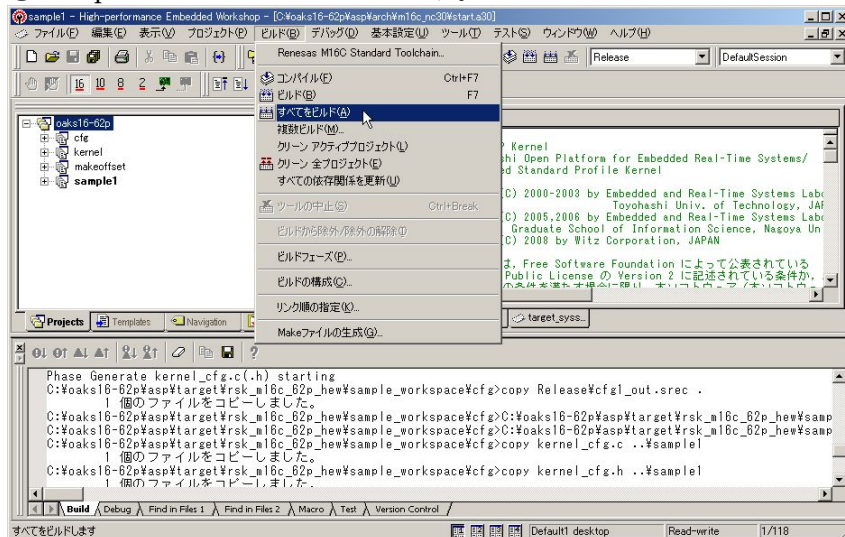
④MCU 設定



MCU 設定:
プロセッサモード: Single-Chip Mode

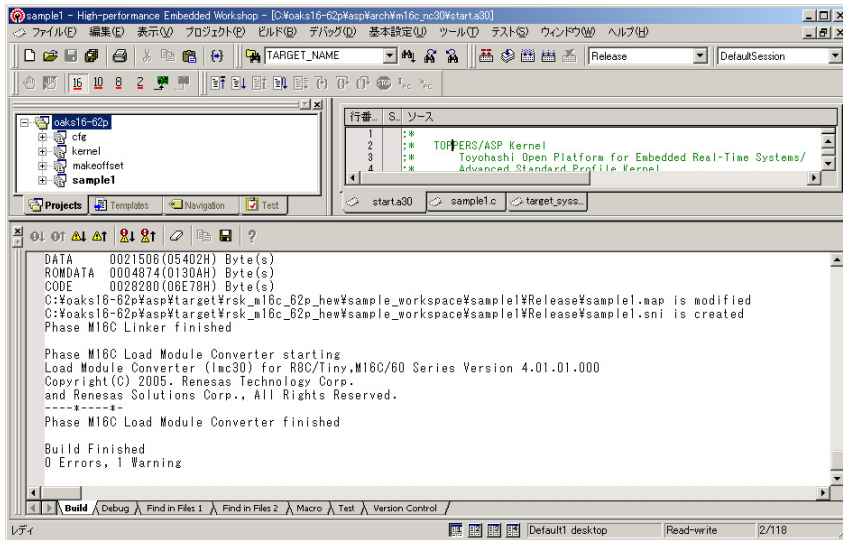
エミュレータ設定が完了しましたら、『OK』をクリックします。

⑤Sample1 プロジェクトをビルドします。



メニュー→ビルド→『すべてをビルド』をクリックします。

しばらくすると、ビルドが完了して、下記のような画面になります。

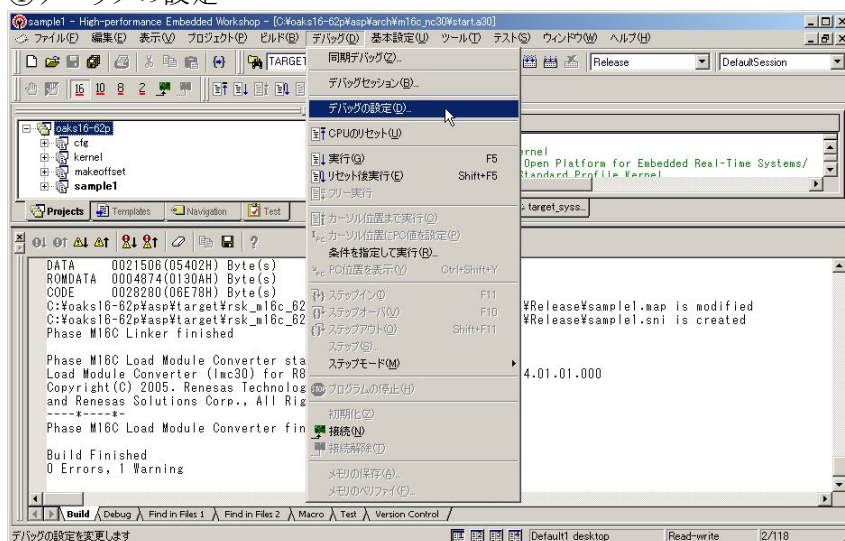


sample1 プロジェクトのビルドが完了した様子。

4.3. デバッグの設定

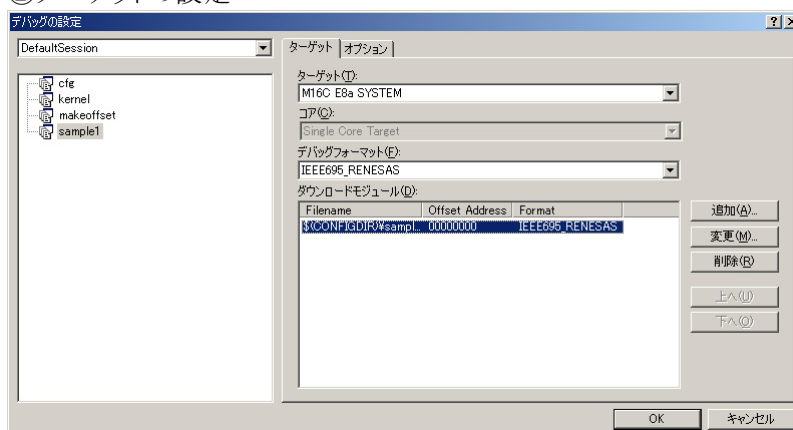
これから、E8a エミュレータを使用してデバッグするため、ターゲットボード OAKS16-62P にダウンロードするファイルを設定します。手順は次のとおりです。

① デバッグの設定



メニュー→デバッグ→『デバッグの設定』をクリックします。

② ターゲットの設定



次のように設定をします。

ターゲット: M16C E8a SYSTEM
デバッグフォーマット: IEEE695_RENESAS
ダウンロードモジュール: sample1.x30

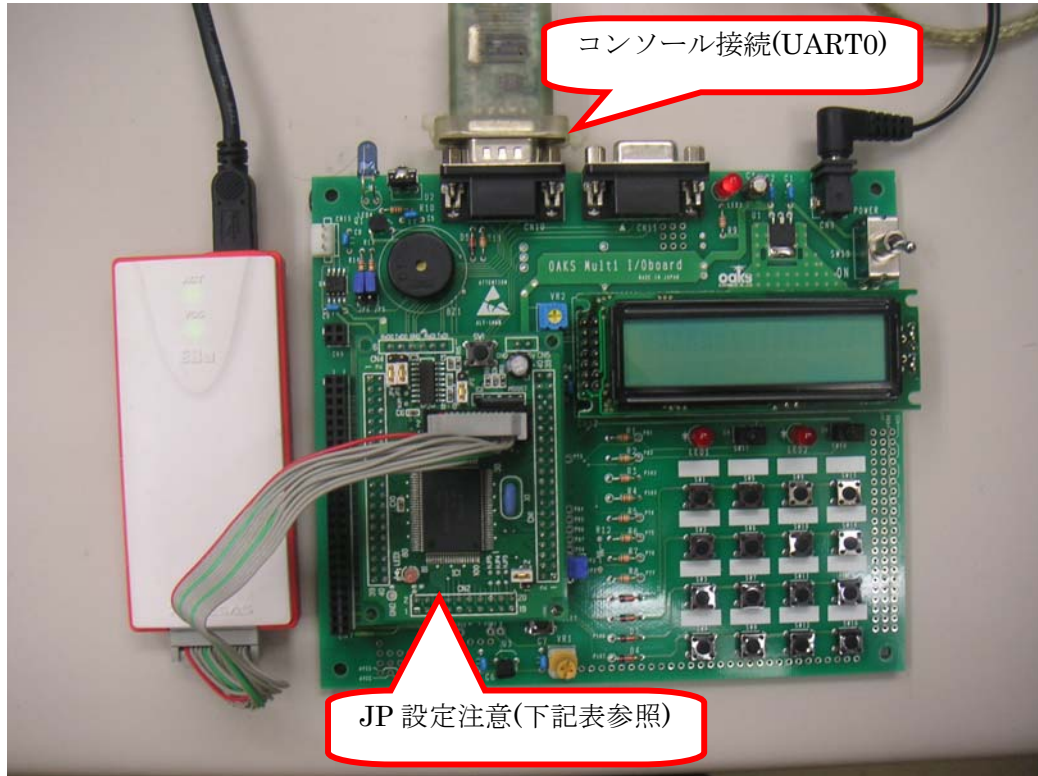
『OK』をクリックします。

(すでに設定済みですので、内容を確認するだけで結構です。)

4.4.ターゲットボードの設定

ターゲットボード OAKS16-62P Boardkit + OAKS Multi I/O Board および E8a エミュレータの接続を行います。

sample1 プログラムは、ターゲットボードよりシリアル I/O ポート入出力がありますので、RS232C ケーブルにて、開発用 PC の COM ポートに接続してください。ハイパーターミナルなどのソフトと通信を行います。

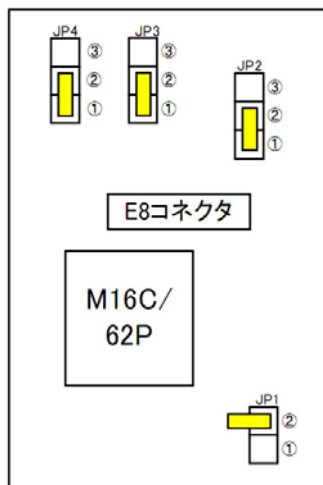


OAKS16-62P Boardkit の JP 設定は、下記のように設定します。

(3) E8使用時

UART2をEXboardのRS232Cで使用する。

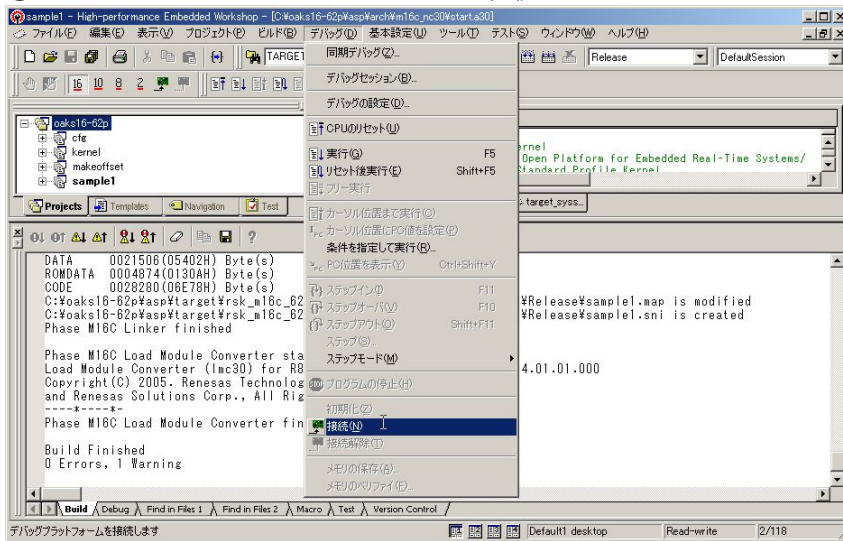
JP1:オープン
JP2:Vcc
JP3:TxD2
JP4:Rx/D2



4.5. デバッグの開始

それでは、接続に間違いがないことを確認しましたら、ターゲットボードの電源を ON して、開発用 PC から E8a エミュレータと通信をしてデバッグを開始します。

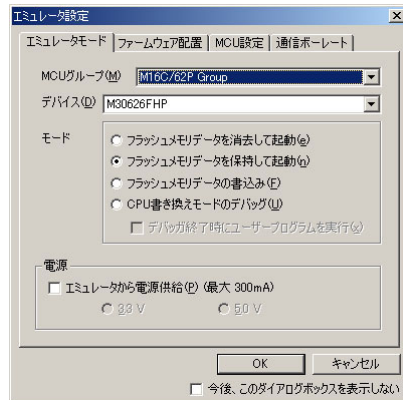
① E8a エミュレータとターゲットボードとの接続



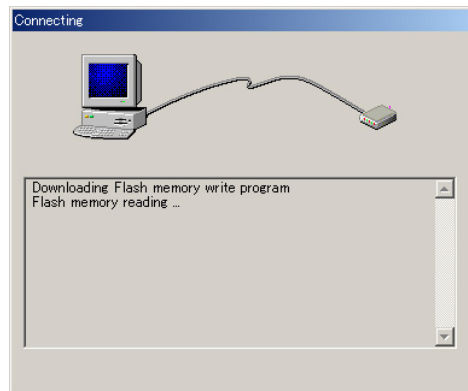
メニュー→デバッガー→『接続』をクリックします。

② エミュレータの設定について

先ほど、エミュレータの設定を行いましたので、特に設定する必要はありません。



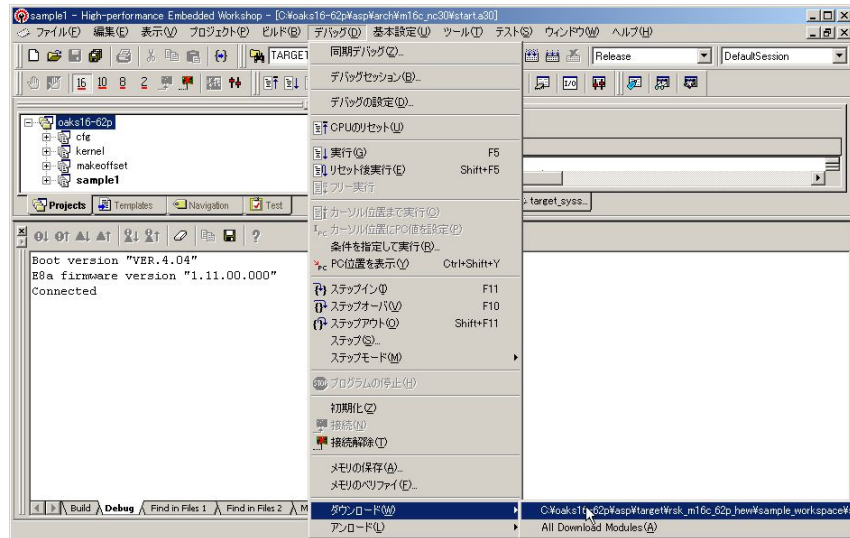
『OK』をクリックします。ターゲットボードと接続を開始します。



接続を開始した様子。

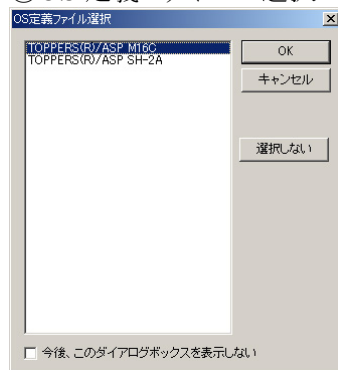
③プログラムのダウンロード

デバッグするプログラムをターゲットボードにダウンロードします。



メニュー→デバッグ→『ダウンロード』をクリックして、ダウンロードするファイルを選択します。

④OS 定義ファイルの選択



TOPPERS/ASP_M16C を選択します。
『OK』をクリックします。

これで、従来のデバッグ手順にてプログラムを実行できますが、sample1 プログラムは、シリアル I/O 入出力を使用するため、プログラムの動作を確認するには、ハイパーターミナルなどの通信ソフトを使用します。

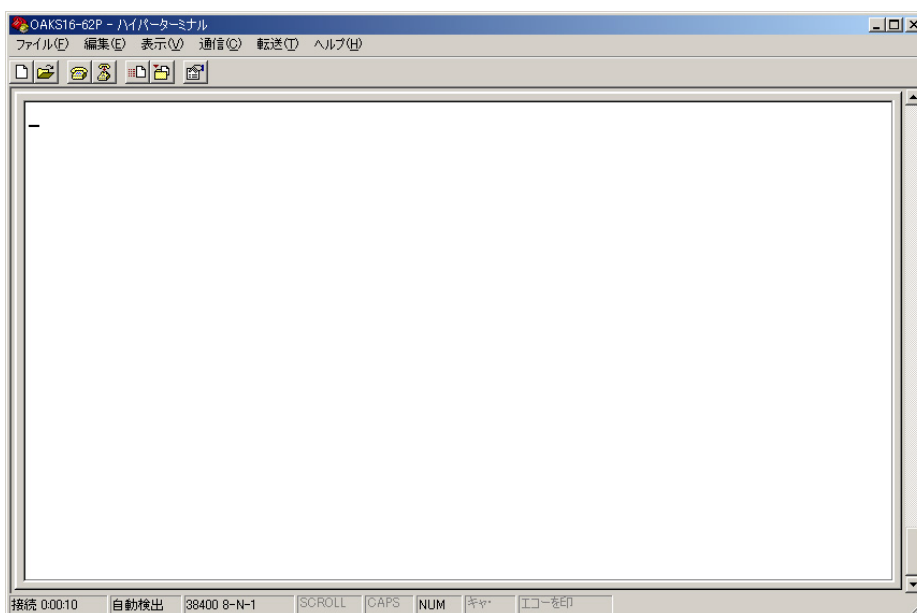
そのため、デバッガでプログラムを実行する前に、ハイパーターミナル通信ソフトで、先に接続をしておきます。

4.6.ハイパーターミナルの設定

それでは、ターゲットボードと通信できるようにするため、ハイパーターミナルを起動します。
ここでは、開発用 PC の COM1 ポートに RS-232C ケーブルで接続していることを前提に説明を致します。

①ハイパーターミナル、ソフトの起動

スタート→プログラム→アクセサリ→通信→『ハイパーターミナル』をクリックします。



ハイパーターミナルが起動した様子。

②新しい接続

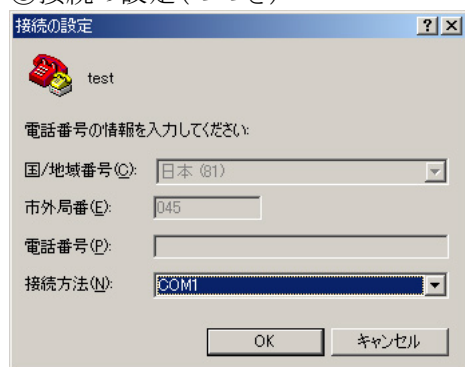


接続設定の名称を入力します。

ここでは、『OAKS16-62P』
と入力して、『OK』をクリックします。

(名称は同じである必要はありません。)

③接続の設定(つづき)



接続方法: COM1
を選択して、『OK』をクリックします。

④ポートの設定



ボーレートの設定: 38400bps
データビット: 8bit
パリティ: なし
ストップビット: 1
フロー制御: なし

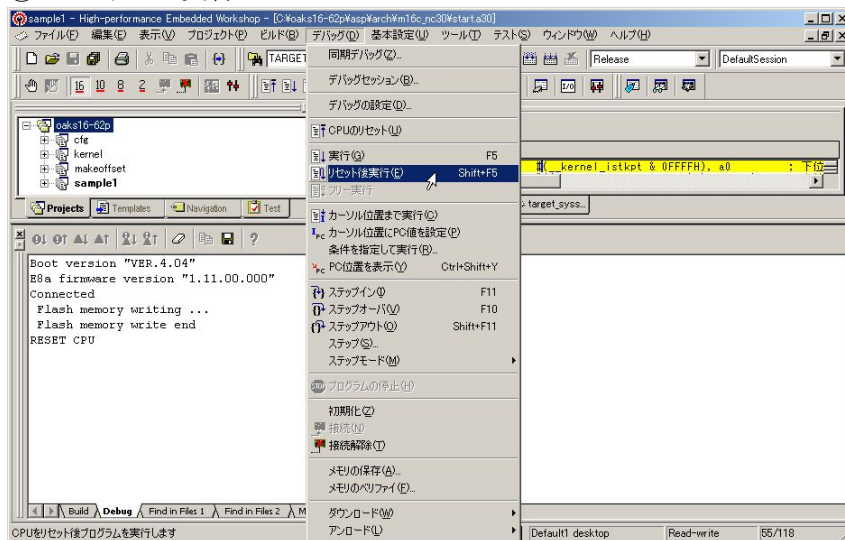
設定が完了しましたら、『OK』をクリックします。

通信が開始されます。

4.7.プログラムの実行

それでは、早速サンプルプログラムを実行してみましょう。手順は次のとおりです。

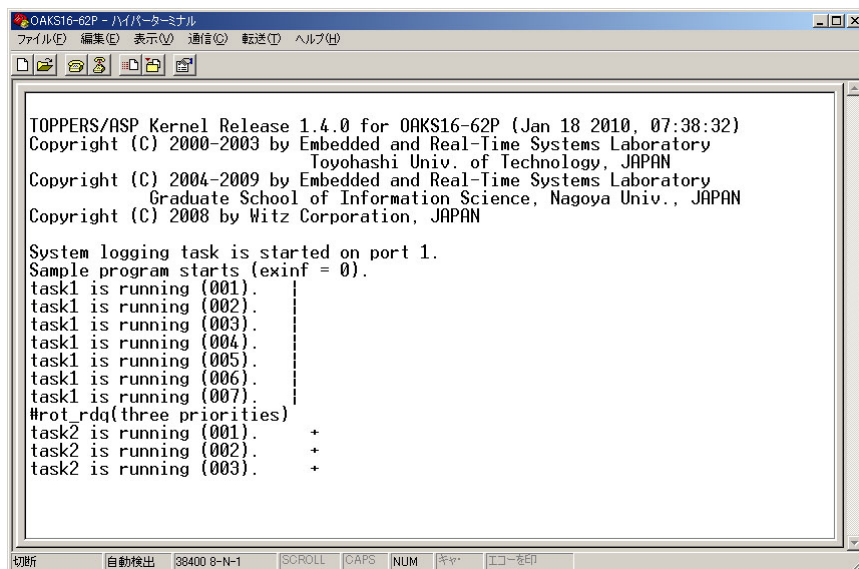
①プログラムの実行



メニュー→デバッグ→『リセット後実行』をクリックします。

ターゲットボードより sample1 プログラムを実行すると、ログが出力されます。ハイパーターミナルの画面を示します。

TOPPERS/ASP カーネルのバージョン表示などがされた後、タスクの実行状況が表示されています。途中で、『r』キーを入力すると、task1 から task2 に切り替わっています。



動作している様子。

『1』、『2』、『3』などのキーを押すと、タスクを切り替えられます。その他にもいろいろとあります。

4.8. サンプルプログラムの動作

先ほどまでの手順で、無事に TOPPERS/ASP カーネルと `sample1` プログラムを動作させることができました。

ここでは、`sample1` プログラムの動作を見てみましょう。

サンプルプログラムの説明は、`sample1.c` ソースコードのコメントにあります。

一部を抜粋します。

```
/*
 * サンプルプログラム(1)の本体
 *
 * ASP カーネルの基本的な動作を確認するためのサンプルプログラム.
 *
 * プログラムの概要:
 *
 * ユーザインタフェースを受け持つメインタスク(タスク ID: MAIN_TASK, 優
 * 先度: MAIN_PRIORITY)と, 3 つの並列実行されるタスク(タスク ID:
 * TASK1~TASK3, 初期優先度: MID_PRIORITY)で構成される. また, 起動周
 * 期が 2 秒の周期ハンドラ(周期ハンドラ ID: CYCHDR1)を用いる.
 *
 * 並列実行されるタスクは, task_loop 回空ループを実行する度に, タスクが
 * 実行中であることをあらわすメッセージを表示する.
 *
 * 周期ハンドラは, 三つの優先度(HIGH_PRIORITY, MID_PRIORITY,
 * LOW_PRIORITY)のレディキューを回転させる. プログラムの起動直後は,
 * 周期ハンドラは停止状態になっている.
 *
 * メインタスクは, シリアル I/O ポートからの文字入力を行い(文字入力を
 * 待っている間は, 並列実行されるタスクが実行されている), 入力された
 * 文字に対応した処理を実行する. 入力された文字と処理の関係は次の通り.
 * Control-C または'Q'が入力されると, プログラムを終了する.
 *
 * '1': 対象タスクを TASK1 に切り換える(初期設定).
 * '2': 対象タスクを TASK2 に切り換える.
 * '3': 対象タスクを TASK3 に切り換える.
----以降省略----
```

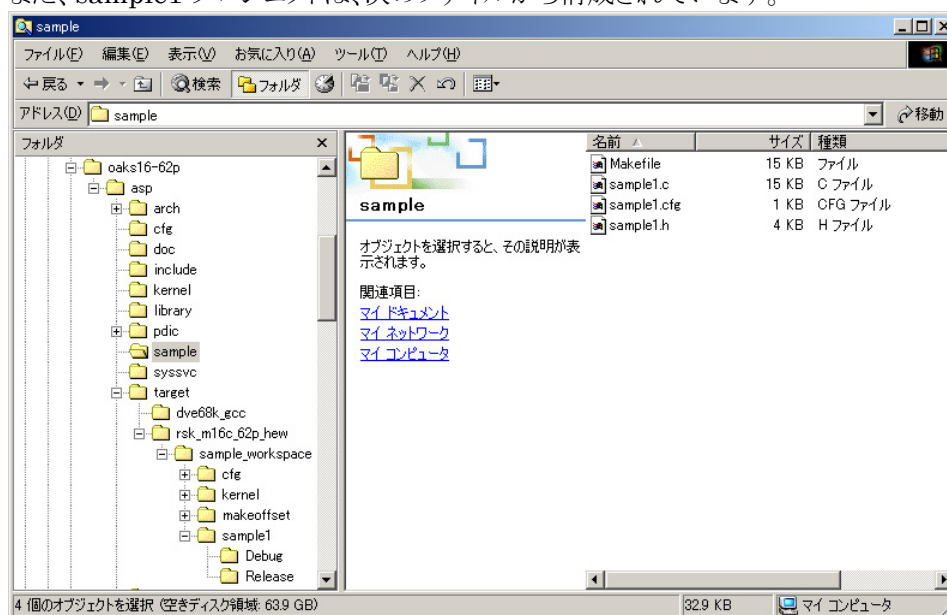
ここまでの説明で、このサンプルプログラムは、4 つのタスクで構成されていることが分かります。

- MAIN_TASK (ユーザーからのキー入力を受け付ける)
- TASK1
- TASK2
- TASK3

詳しくソースコードを見ると、C 言語で最初に実行される `main` 関数が見当たりません。これはどういうことでしょうか。

TOPPERS/ASP カーネルでソフトウェアを作成する場合には、`main` 関数は使用しません。代わりにタスクの定義を行います。タスクの実行のタイミングや優先度など、TOPPERS/ASP カーネルが担当することになります。

また、sample1 プロジェクトは、次のファイルから構成されています。



- sample1.c (ソースコード)
- sample1.h (ヘッダー定義ファイル)
- sample1.cfg (コンフィグレーションファイル)

ここで、特に目新しいものは、sample1.cfg のコンフィグレーションファイルです。

このコンフィグレーションファイルは、タスクの静的生成のために定義するファイルになります。

詳細については、TOPPERSプロジェクトの[TOPPERS 新世代カーネル用コンフィギュレータ仕様書](#)を参照してください。

このように、C 言語での OS を使用しないソフトウェア開発の場合と、TOPPERS/ASP カーネルとでは異なる点があるということを理解していただければ幸いです。

5. TOPPERS/ASP カーネル技術資料

ここでは、TOPPERS/ASP カーネルについて、技術的な内容について説明します。

5.1. フォルダ構成

TOPPERS/ASP カーネルのフォルダについて説明します。
C:\oaks16-62p\asp 以下のフォルダ構成は次のとおりです。

フォルダ名	内容説明
arch	ターゲット依存部の共通部分
cfg	カーネルコンフィギュレータ
doc	ドキュメント
include	アプリケーション向けヘッダファイル
kernel	カーネルソースファイル
library	サポートライブラリソースファイル
pdic	PDIC (デバイスドライバの OS 非依存部分)
sample	サンプルプログラムと Makefile
sysvc	システムサービスヘッダファイル, ソースファイル
target	ターゲット依存部
test	テストプログラム
utils	ユーティリティプログラム

5.2. ドキュメントについて

TOPPERS/ASP カーネルのドキュメントについて説明します。
C:\oaks16-62p\asp\doc にあるファイルになります。

ファイル名	内容説明
user.txt	ユーザーズマニュアル
asp_spec.txt	TOPPERS/ASP カーネルの仕様概要
extension.txt	機能拡張・チューニングガイド
porting.txt	ターゲット依存部 ポーティングガイド
configurator.txt	コンフィギュレータ仕様
design.txt	設計メモ
version.txt	変更履歴

5.3. 参考資料

TOPPERS/ASP カーネルについて、より深く学習するために次のようなドキュメントがあります。
ぜひ一度目を通すことをお勧めいたします。

TOPPERS プロジェクトのホームページよりダウンロードすることができます。

- ・ [TOPPERS新世代カーネル統合仕様書](#)
- ・ [TOPPERS新世代カーネルへのマイグレーションガイド](#)
- ・ [TOPPERS新世代カーネル用コンフィギュレータ関連仕様書](#)

TOPPERS/ASP カーネルは、 μ ITRON4.0 準拠ですので、それらの資料も参考になることでしょう。

6.補足資料

OAKS16-62P boardkitとOAKS Multi I/O Boardを組み合わせて、TOPPERS/ASPカーネルを動作させるための補足資料です。

TOPPERSプロジェクトのTOPPERS/ASPカーネル**RSK-M16C-62P** (ルネサステクノロジ) 簡易パッケージ [asp_rsk_m16c_62p_hew-20091201.lzh](#)をベースに、ルネサステクノロジ製E8aエミュレータ環境にて実行できるようにしたのですが、具体的なソースコードの修正点などについて説明いたします。

6.1.ソースコードの修正について

ソースコードのファイル名:

C:\oaks16-62p\asp\target\rsk_m16c_62p_hew\sectm16c.inc

修正内容:

メモリマップが異なるため、内蔵FlashROMの部分を変更。

DEFAULT_ISTK_TOP:

```
;  
; CODE 領域のセクション  
;  
.section program, CODE  
.org 0A0000H ; 080000H -> 0A0000H に変更  
  
;  
; RO データ領域のセクション  
;  
.section rom_FE, ROMDATA, ALIGN  
.section rom_FO, ROMDATA
```

ソースコードのファイル名:

C:\oaks16-62p\asp\target\rsk_m16c_62p_hew\target_syssvc.h

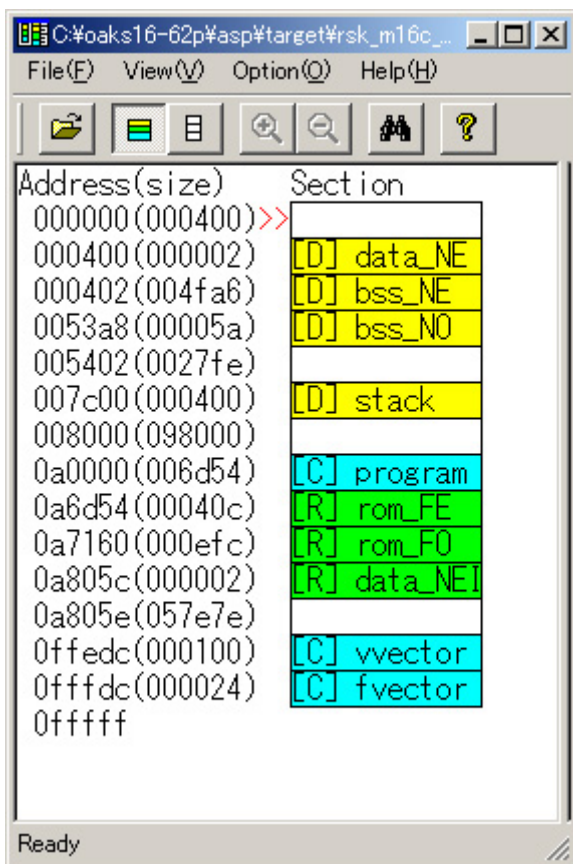
修正内容:

ターゲットシステム名称の設定。

```
/*  
 * 起動メッセージのターゲットシステム名  
 */  
#define TARGET_NAME "OAKS16-62P" /* RSK-M16C-62P ==> OAKS16-62P に変更  
*/
```

6.2.メモリマップの状況

TOPPERS/ASP カーネルと sample1 プロジェクトをビルドしたときのメモリマップは次のとおりです。



The screenshot shows a window titled "C:\oaks16-62p\asp\target\#rsk_m16c_..." with a menu bar (File, View, Option, Help) and a toolbar. The main area displays a memory map with two columns: "Address(size)" and "Section". The sections are color-coded: yellow for data and bss, cyan for program and vectors, and green for ROM and data.

Address(size)	Section
000000(000400)>>	
000400(000002)	[D] data_NE
000402(004fa6)	[D] bss_NE
0053a8(00005a)	[D] bss_NO
005402(0027fe)	
007c00(000400)	[D] stack
008000(098000)	
0a0000(006d54)	[C] program
0a6d54(00040c)	[R] rom_FE
0a7160(000efc)	[R] rom_F0
0a805c(000002)	[R] data_NE1
0a805e(057e7e)	
0ffedc(000100)	[C] vvector
0ffffc(000024)	[C] fvector
0fffff	

OAKS16-62P TOPPERS/ASP ユーザーズマニュアル Rev 1.02

2010年2月初版発行

編集 オークス電子株式会社

発行 オークス電子株式会社

禁無断転載

本説明書の一部又は全部を、当社に断りなく、いかなる形でも転載又は複製することを堅くお断りします。