

---

# OAKS32-LANBoard 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。

Microsoft, MS 及びMS-DOS は、米国Microsoft Corporation の登録商標です。  
Windows95, Windows98 は、米国Microsoft Corporation の商標です。  
IBM 及びPC/AT は、米国International Business Machines Corporation の登録商標です。  
Pentium は、米国Intel Corporation の商標です。  
Adobe, Acrobat は、Adobe Systems Incorporated(アドビシステムズ社)の商標です。

－はじめに－

このたびは**OAKS32-LANBoard**をお買い上げいただき誠にありがとうございます。  
このマニュアルは、**OAKS32-LANBoard** に含まれるハードウェアおよびソフトウェアのセットアップ方法、  
使用上の注意点について述べたものです。各ソフトウェアの詳細な操作方法については CDROM 内の各マ  
ニュアルを参照ください。



－本マニュアル内での語句に関して－

特に断りのない限りNC308WA（オークス版）をNC308WA、KD3083（オークス版）をKD3083、TM（オークス版）をTMと呼んでいます。

－本マニュアル内での動作環境について－

ソフトウェア（ツール、サンプルプログラム）は全てWindows2000の環境で動作させたものとしています。

<b>1. OAKS32-LANBOARD の概要</b> .....	<b>8</b>
1.1 製品パッケージの内容.....	8
1.2 CD-ROM の内容.....	8
1.3 動作環境.....	9
1.4 OAKS32-LANBoard の特徴.....	10
1.4.1 OAKS32-LANBoard で出来ること.....	11
1.4.2 リアルタイム OS (MR308 オークス版).....	11
1.4.3 TCP/IP プロトコルスタック (TCP308 オークス版).....	11
1.4.4 OAKS32-LANBoard でのユーザプログラムの実機評価について.....	12
1.5 OAKS32-LANBoard のドキュメント構成.....	13
<b>2. 開発環境を整えよう</b> .....	<b>15</b>
2.1 NC308WA のセットアップ.....	15
2.1.1 NC308WA について.....	15
2.1.2 NC308WA のインストール.....	16
2.1.3 NC308WA の動作確認.....	17
2.1.4 NC308WA のアンインストール.....	18
2.2 リモートデバッガ (KD3083) のセットアップ.....	19
2.2.1 KD3083 について.....	19
2.2.2 KD3083 のインストール.....	20
2.2.3 KD3083 の起動.....	20
2.2.4 KD3083 の終了.....	22
2.2.5 KD3083 のアンインストール.....	23
2.2.6 KD3083 のエラー対策.....	23
2.2.7 ソースファイルを使用したデバッグを行うために.....	24
2.3 内蔵フラッシュメモリ書き込みプログラム (Flash Starter) のセットアップ.....	25
2.3.1 Flash Starter について.....	25
2.3.2 Flash Starter のインストール.....	26

2.3.3 Flash Starter の起動.....	26
2.3.4 KD3083 モニタプログラム機械語ファイルの書き込み方 .....	28
2.3.5 Flash Sterter のアンインストール .....	29
<b>2.4 TM (Tool Manager) のセットアップ.....</b>	<b>30</b>
2.4.1 TM について.....	30
2.4.2 TM のインストール.....	30
2.4.3 TM の起動.....	30
2.4.4 TM のカスタマイズ.....	31
2.4.5 TM のアンインストール.....	34
<b>2.5 拡張 LED ボードの組み立て.....</b>	<b>35</b>
<b>3. サンプルプログラムを動かしてみよう .....</b>	<b>37</b>
3.1 サンプルプログラムの内容 .....	37
3.2 LED テストプログラムを動かす .....	38
3.2.1 LED テストプログラムの動作.....	38
3.2.2 LED テストプログラムのファイル構成 .....	38
3.2.3 プログラムのビルドから動作させるまでの手順.....	39
<b>4 MR308 を使ったプログラム開発.....</b>	<b>43</b>
4.1 MR308 の特徴.....	43
4.2 MR308 で使用するファイル.....	45
4.3 コンパイル時に必要なオプション .....	45
4.4 TM (Tool Manager) でプロジェクトを作成する手順.....	46
4.5 MR308 のサンプルプログラム .....	56
4.5.1 周期起動ハンドラのサンプルプログラム .....	56
4.5.2 イベントフラグのサンプルプログラム .....	58
4.5.2 イベントフラグのサンプルプログラム .....	59
4.5.3 時間管理機能 (タスク遅延) によるサンプルプログラム .....	61

<b>5 STCP308 (オークス版) を使ったプログラム開発 .....</b>	<b>63</b>
5.1 STCP308 について .....	63
5.2 STCP308 と OAKS32-LANBoard の H/W .....	64
5.2 STCP308 の機能について .....	65
5.3 STCP308 で使用するファイル .....	66
5.4 コンパイル時に必要なオプション .....	67
5.5 TM (Tool Manager) でプログラムをビルドするまでの手順 .....	68
5.6 STCP308 のサンプルプログラム .....	79
5.6.1 ping を試してみよう .....	79
5.6.2 ARP コマンドで MAC アドレスを調べてみよう .....	83
5.6.3 telnet のサンプルプログラム .....	85
<b>6 制限事項 .....</b>	<b>88</b>
6.1 端子の使用に関する制限事項 .....	88
6.2 周辺機能の使用に関する制限事項 .....	89
6.3 メモリの使用に関する制限事項 .....	90
6.4 レジスタ操作に関する制限事項 .....	92
6.5 割り込みに関する制限事項 .....	95
6.6 命令に関する制限事項 .....	96
6.7 KD3083 に関する制限事項 .....	97
6.7.1 ブレークポイントに関する制限事項 .....	97
6.7.2 ストップモード、ウェイトモードに関する制限事項 .....	97
6.7.3 ユーザプログラムのリアルタイム性について .....	97
6.7.4 例外的なステップ実行について .....	98
<b>7 保証とサポート .....</b>	<b>99</b>

7.1 保証について.....	99
7.2 技術サポートについて.....	99
付録.....	100
部品表.....	101
寸法図.....	103
改訂記録.....	104

# 1. OAKS32-LANBoard の概要

## 1.1 製品パッケージの内容

OAKS32-LANBoardパッケージの包装内容を示します。開封時に包装内容をご確認下さい。

表1-1 OAKS32-LANBoardパッケージ包装内容一覧表

製品名	数量	備考
OAKS32LAN-Board	1枚	部品実装済み TCP/IPプロトコルスタック+リアルタイムOS+リモートデバッグモニタプログラム書き込み済み 100mm×72mm
ジャンパプラグ	1個	OAKS32LAN-BoardのJP1コネクタ接続用
拡張ボード一式 (要組み立て)	プリント基板	1枚 配線済み 50mm×10mm
	緑色単体LED	8個
	抵抗	8個 1kΩ
	コネクタ	1個
CD-ROM	1枚	マニュアル類 開発ツール サンプルプログラム

電源は本キットに含まれておりません。本キットには必ず5V安定化電源をお使いください。オークス電子で電源、ケーブルセットを販売しております。

- 電源・ケーブルセット (5V安定化電源、RS232Cストレートケーブル、LANクロスケーブル) 3,000円

## 1.2 CD-ROM の内容

CD-ROM にはプログラム開発に必要なソフトウェア製品、マニュアルなどが含まれています。

表1-2 OAKS32-LANBoard CD-ROM内容一覧表

ディレクトリ	内容
Doc_OAKS32	本マニュアル
Doc_M32C-83	M32C/83シリーズに関するソフトウェア/ハードウェアドキュメント類
Tools	開発ツールとそのマニュアル類
Sample	サンプルプログラム
STCP308	STCP308オークス版 M32C/83シリーズ用TCP/IP STCP308ライブラリリリースノート STCP308+MR308+KD3038モニタプログラムの機械語ファイル (モトローラフォーマットファイル)
MR308	MR308オークス版 M32C/83シリーズ用リアルタイムOS MR308リリースノート ※MR308のみの機械語ファイルは含まれません。

### 1.3 動作環境

以下の物については別途ご用意下さい。

- ・ホストパーソナルコンピュータ
- ・RS232Cケーブル：9pinオスメス型ストレートケーブル
- ・10-BASET LANケーブル（OAKS32-LANBoardをLANに接続する場合に必要です）

表1-3にパーソナルコンピュータのスペックを示します。

表1-3 ホストパーソナルコンピュータのスペック

PC	IBM PC/ATシリーズおよびその互換機
OS	Microsoft Windows95/98/ME/NT/2000/XP
ブラウザ	Microsoft Internet Explorer4.0以上（統合化開発環境：TM使用時に必要です）
CPU	Pentium II 233MHz 以上を推奨
メモリ	128Mバイト以上を推奨
CDドライブ	CD-ROM読み込み可能なドライブ
ポート	シリアルポート 1チャンネル
インターフェース	LANインターフェース(OAKS32-LANBoardをLANに接続する場合に必要です)

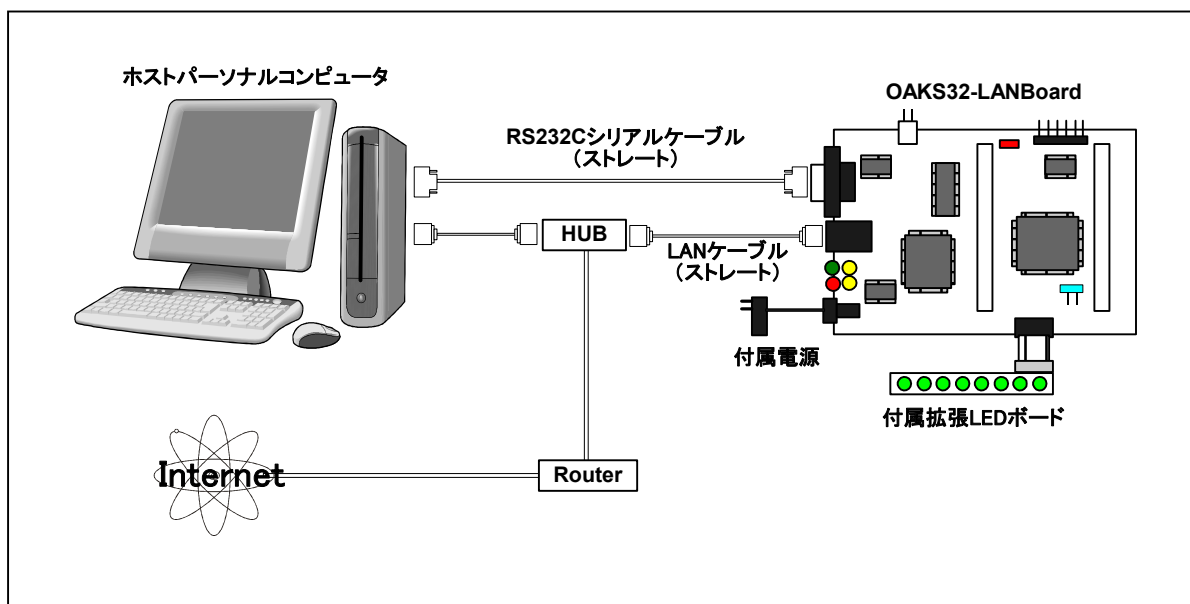


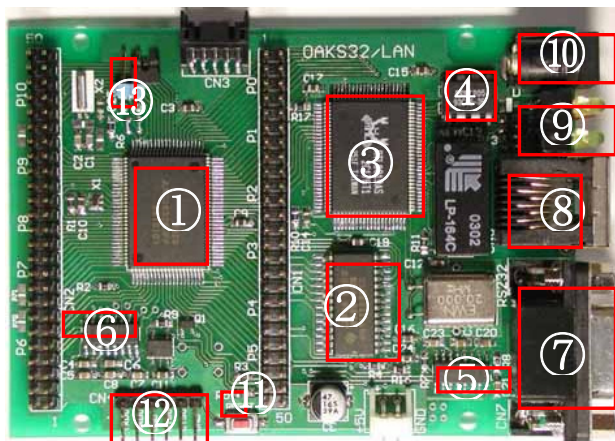
図 1-1 OAKS32-LANBoard構成例

※OAKS32-LANBoardからHUBを経由せずに直接パーソナルコンピュータへ接続する場合は、クロスLANケーブルをお使いください。

※電源は必ず5V安定化電源をお使いください。

## 1.4 OAKS32-LANBoard の特徴

ルネサス テクノロジ社製16ビットシングルチップマイクロコンピュータ M16Cファミリ M32C/83 シリーズのフラッシュメモリ内蔵版 M30833FJFPを搭載したLANボードです。  
内蔵フラッシュメモリにはあらかじめ、リモートデバッガモニタプログラム+リアルタイムOS+TCP/IPプロトコルスタックの機械語ファイルが書き込まれています。



OAKS32-LANBoard

表2-1 OAKS32-LANBoardの構成

名称	型番	内容	図中番号
CPU	M30833FJFP	ルネサス テクノロジ社製16ビットシングルチップマイクロコンピュータ M16Cファミリ M32C/83 シリーズ (100ピン版) のフラッシュメモリ内蔵版です。	①
拡張メモリ	M5M417800	DRAM 2 Mbyteを搭載しています。	②
Ethernet コントローラ	RTL8019AS	TCP/IP通信に使用します。	③
EEPROM	93C46-3P	ボードごとにユニークなMACアドレスがあらかじめ書き込まれています。	④
RS232C ドライバ	MAX232A	⑤はホストマシンとの通信に使用します。⑥は拡張機能としてユーザーに開放されています。	⑤、⑥
RS232C コネクタ		RS232Cケーブルを接続します。	⑦
RJ45コネクタ		LANケーブルを接続します。	⑧
LED0~3		LED0:電源投入の間点灯します。(電源回路と接続) LED2:リンクしている場合点灯します。(RTL8019ASと接続) LED3:データを受信している間点灯します。(RTL8019ASと接続) LED4:データを送信している間点灯します。(RTL8019ASと接続)	⑨
電源コネクタ		電源を接続します。	⑩
リセットSW		CPUをリセットします。	⑪
拡張ボード コネクタ		LED拡張ボードを接続します。 CPUのポート0端子と接続しています。	⑫
JP1コネクタ		オープン:リセット後、リセットベクタアドレスより命令を読み出します。(KD3038、ユーザプログラム起動時) ショート:リセット後、フラッシュメモリ書き換えブートプログラムが起動します。(FLASHSTA起動時)	⑬

※ H/Wについての詳細は「付録 OAKS32LanBoard H/W仕様」を参照してください。

※ CPU以外の部品などについては相当品となる場合があります。

#### 1.4.1 OAKS32-LANBoard で出来ること

OAKS32-LANBoardで以下の事ができます。

- M32C/83シリーズ マイコンの評価、学習
- M32C/83シリーズ用 リアルタイムOS「MR308」の評価、学習
- M32C/83シリーズ用 TCP/IPプロトコルスタック「STCP308」の評価、学習

#### 1.4.2 リアルタイム OS (MR308 オークス版)

リアルタイムOS (MR308オークス版) はルネサステクノロジ社製ワンチップマイコンM16ファミリM32C/83グループ用に開発されたリアルタイムOSの機能限定版です。ルネサステクノロジ社製の正規版ではソースファイルで提供されますが、MR308オークス版ではKD308モニタプログラム、TCP308も含めた機械語ファイル (stcp308\_mr308mon.mot) で提供しています。

※MR308を使用する上で、ユーザプログラムに制限事項が発生します。  
詳細は「第6章 制限事項」を参照ください。

#### 1.4.3 TCP/IP プロトコルスタック (TCP308 オークス版)

TCP/IPプロトコルスタック (STCP308オークス版) はルネサステクノロジ社製ワンチップマイコンM16ファミリM32C/83グループ用に開発されたネットワークソフトウェアライブラリの機能限定版です。ルネサステクノロジ社製の正規版ではライブラリソースで提供されますが、TCP308オークス版ではKD308モニタプログラム、MR308も含めた機械語ファイル (stcp308\_mr308mon.mot) で提供しています。TCP308オークス版の各ライブラリは、リアルタイムOS (MR308) と共に動作します。

※STCP308を使用する上で、ユーザプログラムに制限事項が発生します。  
詳細は「第6章 制限事項」を参照ください。

#### 1.4.4 OAKS32-LANBoard でのユーザプログラムの実機評価について

OAKS32-LANBoardでは、ユーザプログラムをデバッガ（KD3083）で評価後、ボードをリセットするだけで実機動作（デバッガを介さないで動作させること）の確認をすることができます。

- ① KD3083を起動し、ユーザプログラムをロードします。  
（フラッシュメモリに書き込まれます）
- ② ユーザプログラムの評価後、KD3083を終了してボードをリセットします。
- ③ リセット後、約5秒後にユーザプログラムが実行されます。

リセット後の5秒間はKD3083の起動コマンド待ちです。この間にKD3083起動のコマンドが送られなかった場合はユーザプログラム領域の先頭アドレス（8000H）より実行します。（ユーザプログラム領域になにも書き込まれていない場合はKD3083起動のコマンド待ちのままとなります）

KD3083を一度終了したあとに再びKD3083を起動する場合、あるいはユーザプログラムを実行する場合はOAKS32-LANBoardをリセットしてください。

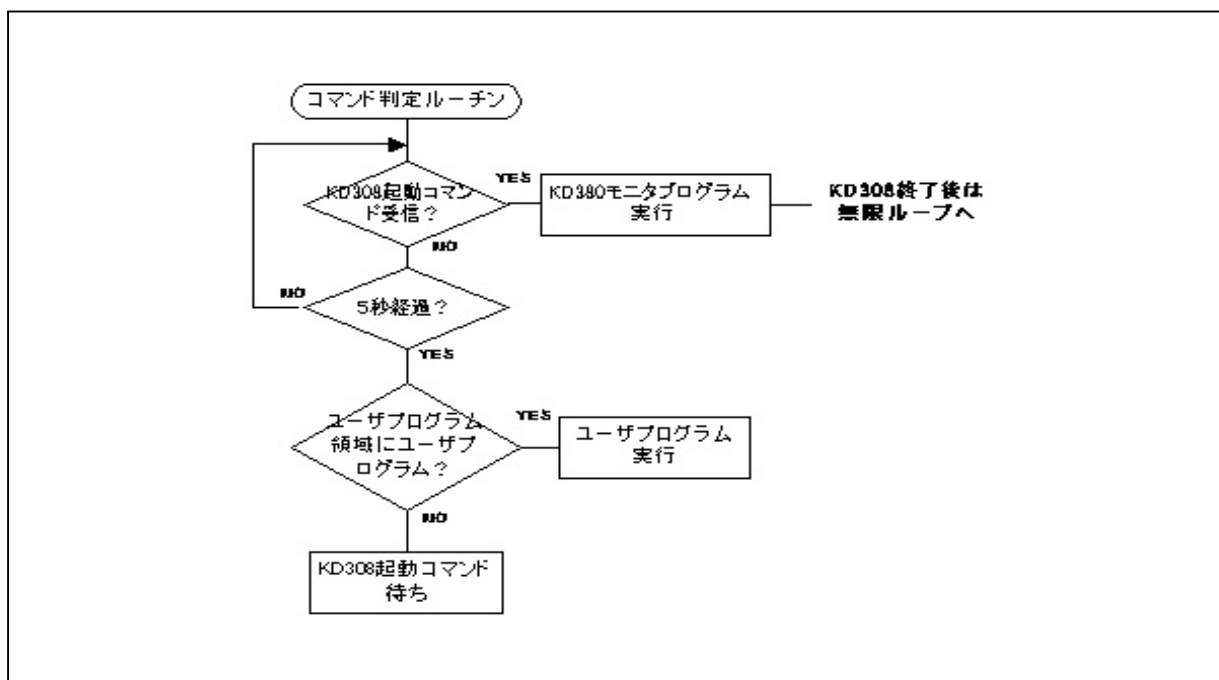


図 1-2 OAKS32-LANBoardのリセット後の動作

## 1.5 OAKS32-LANBoard のドキュメント構成

OAKS32-LANBoardのドキュメント構成は以下のとおりです。

### OAKS32-LANBoardに関するマニュアル

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
Doc_Oaks32	MAN_OAKS32-LANBoard.pdf	OAKS32-LANBoard ユーザーズマニュアル	本マニュアル 最初にお読みください

### CPU-M30833FJFPのH/W、S/Wに関するデータシート、マニュアル類

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
Doc_M32C-83¥DATA_S	83jds.pdf	M32C/83グループ データシート	主にCPUのH/W構成について
Doc_M32C-83¥USERS_M	80jum.pdf	M32C/80グループ ユーザーズマニュアル	主にCPUの使い方について
Doc_M32C-83¥SOFT_M	j32c80sm.pdf	M32C/80シリーズ ソフトウェアマニュアル (暫定版)	主にアセンブラ命令の内容、アドレッシングなど
	80jism.pdf	M16C/80シリーズ ソフトウェアマニュアル	主にアセンブラ命令の内容、アドレッシングなど
Doc_M32C-83¥PROG_M	80JASM.PDF	M16C/80シリーズ アセンブリ言語編 プログラミングマニュアル	アセンブラでプログラムを書くために必要な知識について、(CPU内部、命令、プログラムの書き方等)
	80jc.pdf	M16C/80シリーズ C言語編 プログラミングマニュアル	C言語入門、ROM化の方法、リアルタイムOSを使用した場合のプログラミングについてなど
Doc_M32C-83¥APPLI_N	80jsp.pdf	M16C/80シリーズ アプリケーションノート 「参考プログラム集」	RAMクリア、データブロック転送、文字列の扱い、演算など

★他シリーズのマニュアルは参考としてください。

### Ethernet コントローラ-RTL8019AS のデータシート

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
Doc_Oaks32¥Device	new-8019as.pdf	リアルテック社「RTL8019AS」	Ethernetコントローラのデータシート

### コンパイラに関するマニュアル

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
Tools¥Nc308wa¥MANUAL	AS308UJ.PDF	M16C/80シリーズ リロケータブル アセンブラ 「AS308」ユーザーズマニュアル	操作方法、入出力ファイル、オプション、エラー、ワーニング、指示命令など
	NC308UJ.PDF	M16C/80シリーズ Cコンパイラ「NC308」 ユーザーズマニュアル	操作方法、入出力ファイル、オプション、エラー、ワーニング、指示命令など
Tools¥GNU¥DOC			GNUのGPL, LGPLに関するドキュメント
Tools¥RNOTE	rnote_nc308.pdf	NC308オックス版 リリースノート	NC308オックス版の機能制限について

### TM（統合化開発環境）に関するマニュアル

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
Tools¥TM¥MANUAL	TMUJ.PDF	統合化開発環境 「TM」 ユーザーズマニュアル	操作方法、チュートリアルなど
Tools¥RNOTE	rnote_tm.pdf	TMオークス版 リリースノート	TMオークス版の機能制限について

### Flash Starter に関するマニュアル

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
Tools¥FlashStart	rjj10j0142_m3a0806um.pdf	「M16C Flash Starter」 ユーザーズマニュアル	操作方法

### MR308に関するマニュアル

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
MR308¥doc	mr308rj.pdf	M3T-MR308 リファレンスマニュアル	リアルタイムOSのシステムコールリファレンス
	mr308uj.pdf	M3T-MR308 ユーザーズマニュアル	アプリケーション作成の手引き、サンプルなど
	mr308subj.pdf	MR308 オークス版リリースノート	MR308オークス版の機能制限について

### STCP308に関するマニュアル

CD-ROMのディレクトリ	ファイル名	マニュアル名	内容
STCP308¥doc	stcp308uj.pdf	STCP308 ユーザーズマニュアル	TCP/IP 基礎知識、API使用、用語の定義など
	stcp308subj.pdf	STCP308 オークス版リリースノート	STCP308オークス版の機能制限について

※KD3083の操作マニュアルは画面上のヘルプのみになります。

## 2. 開発環境を整えよう

OAKS32-LANBoardの開発環境を整えましょう。各ソフトウェアの詳細な操作方法は、マニュアルを参照してください。

### 2.1 NC308WA のセットアップ

#### 2.1.1 NC308WA について

NC308WAとは、M16Cファミリ、M32C/83シリーズ用コンパイラの名称です。NC308WAオークスは正規版（ルネサス テクノロジ株式会社製）に対し、一部機能制限があります。主な制限としては、プログラムの最適化機能が使用できません。したがって最適化機能を指定するようなアセンブラ/コンパイルオプションは使用できません。機能制限の詳細については本キットのCD-ROMに収められているリリースノート（Tools/Rnote/rnote\_nc308.pdf）を参照して下さい。

NC308WAオークス版は正規版に対して上位互換性があり、NC308WAオークス版で開発されたプログラムは、正規版でも同様にコンパイルできます。

## 2.1.2 NC308WA のインストール

他のNC308WA(体験版など)がインストールされているコンピュータに、本製品をインストールする場合は、必ずすでにインストールされているNC308WAをアンインストールした後で、本キットのNC308WAをインストールしてください。

(特に、同じバージョンのNC308WAを同じコンピュータにインストールすることは避けてください。)

- ① CD-ROM の¥Tools¥Nc308wa¥W95J ディレクトリ内の SETUP.EXE を実行します。
- ② インストール画面の指示にしたがってインストールします。
- ③ インストール時にAUTOEXEC.BAT を変更する画面が表示されます。

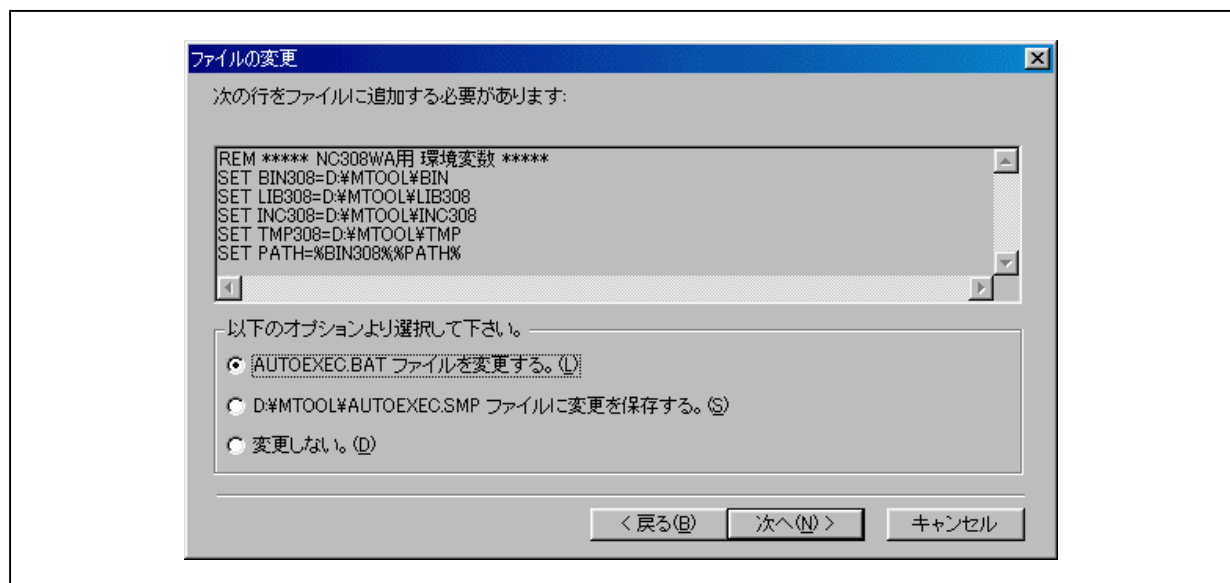


図2-1 環境変数設定ウインドウ

「変更しない」をチェックした場合、下記の内容をsysedit等でAUTOEXEC.BAT に追加するようにしてください。

```
REM ***** NC308WA用 環境変数 *****
SET BIN308= (インストールディレクトリ) ¥BIN
SET LIB308= (インストールディレクトリ) ¥LIB308
SET INC308= (インストールディレクトリ) ¥INC308
SET TMP308= (インストールディレクトリ) ¥TMP
SET PATH=%BIN308%;%PATH%
```

- ④ インストール後、コンピュータを再起動してください。

### 2.1.3 NC308WA の動作確認

NC308WAが動作することを確認します。コマンドプロンプト（MS-DOSプロンプト）でバージョン情報のコマンドを実行します。（NC308WAはMicrosoft Windowsのコマンドプロンプト（MS-DOSプロンプト）か、TM（Tool Manager）より起動できます。）

Windowsの「スタート」-「プログラム」-「アクセサリ」-「コマンドプロンプト」を起動します。

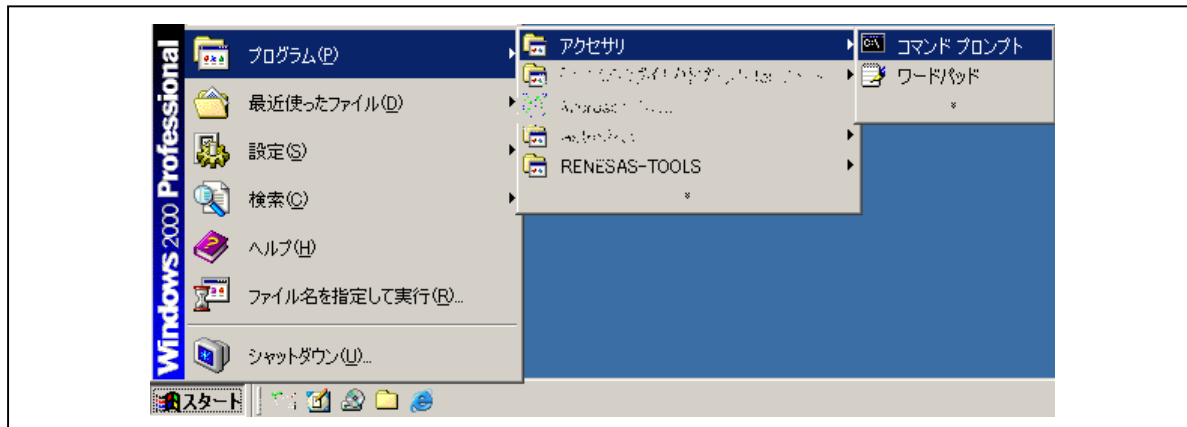


図2-2 Windowsコマンドプロンプトの起動

「nc308 -v」（nc308と-vの間にスペースが入ります）と入力して「Enter」キーを押してください。

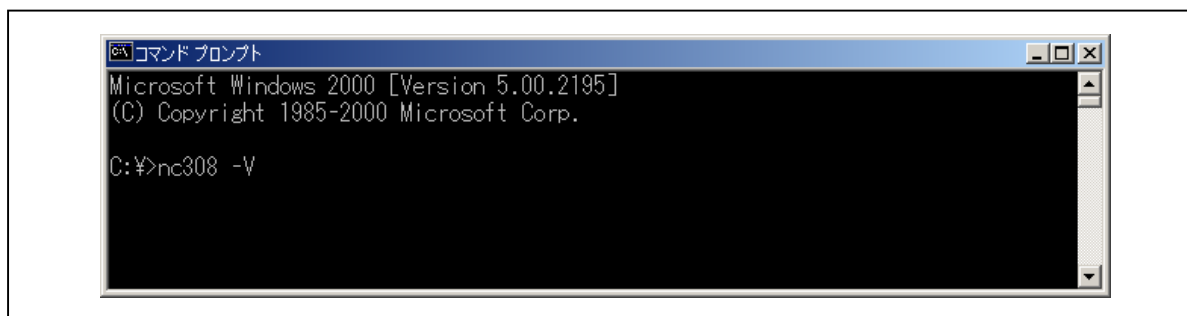


図2-3 NC308WAの起動（バージョン情報オプション）

NC308WAのバージョンとメッセージが以下のように表示されます。

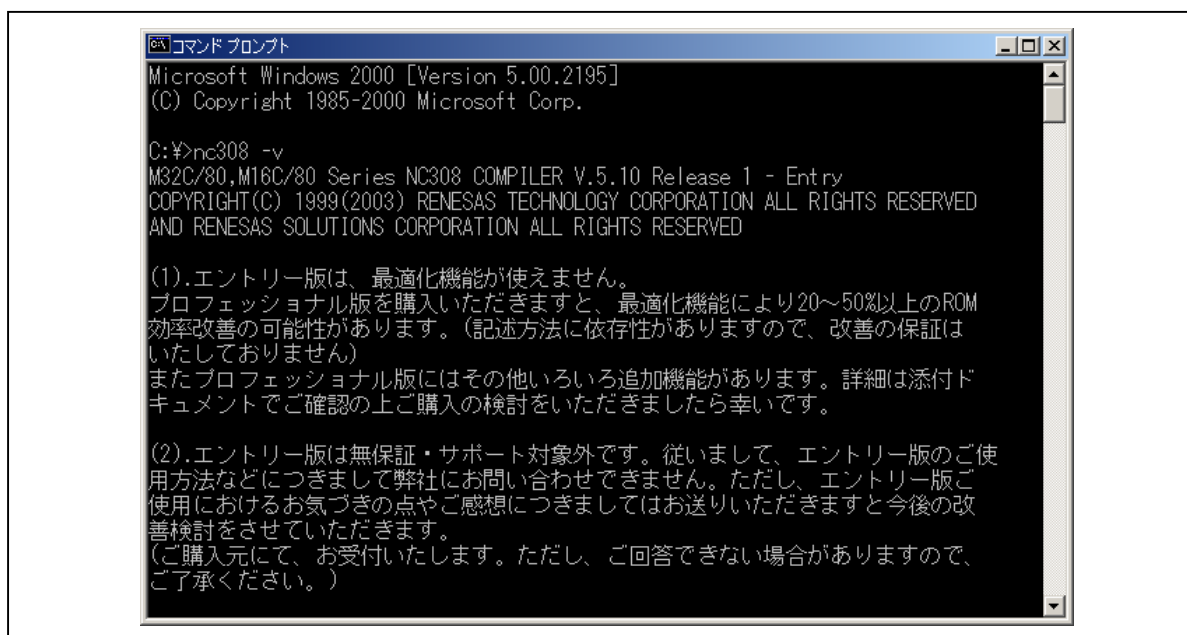


図2-4 NC308WAのバージョン情報とメッセージ

● **NC308WAが起動しない場合**

コマンドパス (PATH) が正しく設定されているか確認してください。

● **バージョン番号が表示されない場合**

BIN30に指定したディレクトリに NC308.EXE が正しく展開されているか確認してください。

#### **2.1.4 NC308WA のアンインストール**

NC308WAをアンインストールするには [スタート]—[設定]— [コントロールパネル] を開き、 [アプリケーションの追加、削除] をクリックします。プログラムリストから [NC308WA V. 5. 10 Release 1] を選択し、 [追加と削除] ボタンをクリックします。アンインストールウィンドウが開き、NC308WAがアンインストールされます。

## 2.2 リモートデバッガ (KD3083) のセットアップ

### 2.2.1 KD3083 について

リモートデバッガ KD3083は、ホストPC側のプログラム (KD3083.exe) と、マイコン内蔵のフラッシュメモリに書き込まれているモニタプログラム (stcp308\_mr308mon.mot : モトローラS形式機械語ファイル) の2つのプログラムで構成されます。

ホストPC側のプログラムとモニタプログラムの通信が確立し、ホストPC側よりユーザプログラムロードコマンドを実行すると、ホストPCよりOAKS32-LANBoardへユーザプログラムがシリアルで送信され、モニタプログラムがフラッシュメモリへ書き込みます。モニタプログラムはホストパーソナルコンピュータ側のプログラムから送信されるコマンドに従って、ユーザプログラムを実行したり、CPUの状態を返信します。このように2つのプログラムが通信を行うことで、特別なH/Wを必要とせずに高機能なデバッグ環境を提供します。

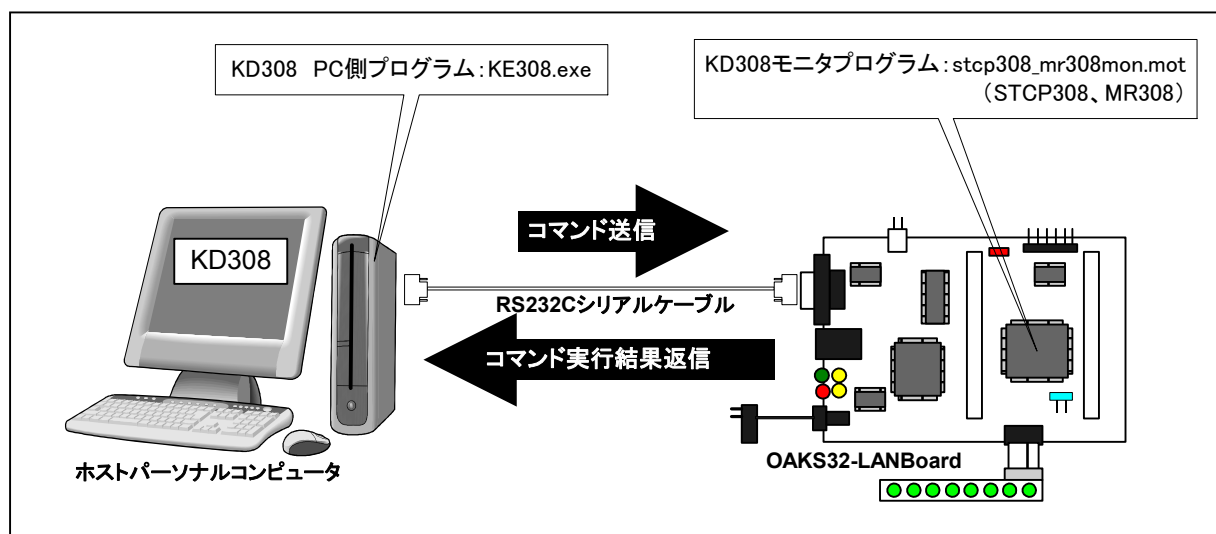


図2-3 OAKS32-LANBoard デバッガ構成図

※KD3083を使用する上で、ユーザプログラムに制限事項が発生します。  
詳細は「第7章 制限事項」を参照ください。

## 2.2.2 KD3083 のインストール

- ①CD-ROMの¥TOOL¥Kd3083 ディレクトリ内の kd3083v300r1\_j.exe を実行します。
- ②インストール画面の指示にしたがってインストールします。

## 2.2.3 KD3083 の起動

KD3083を起動するには、Windowsのスタートメニューから [プログラム] – [RENESAS-TOOLS] – [KD3083 V3.00 Release1] – [KD3083]をクリックします。

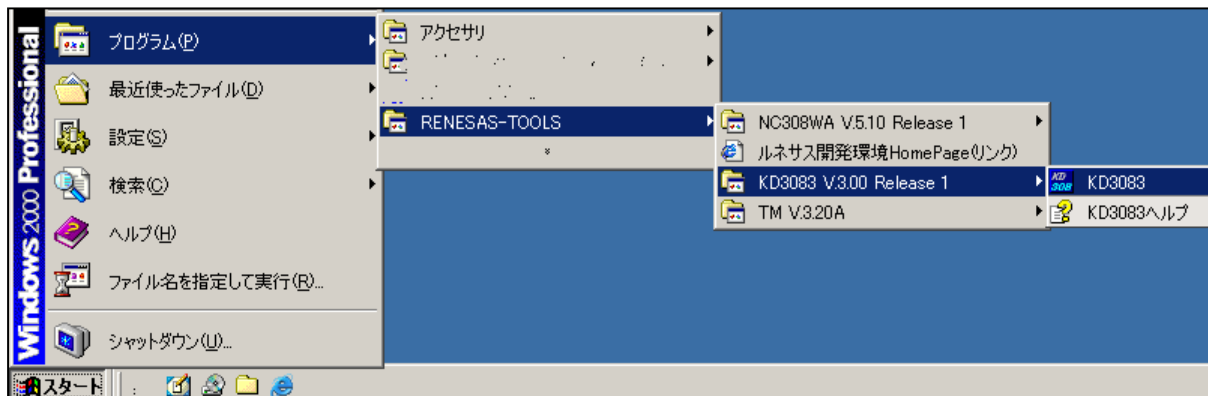


図 2-4 KD3083の起動

KD3083が起動すると、Initウィンドウがオープンします。この時点ではまだボード側のモニタプログラムとは通信していません。

インストール後の初めの起動時にMCUファイルの指定が要求されます。[Refer] ボタンをクリックして” M30835FJ.MCU” ファイルを指定してください。

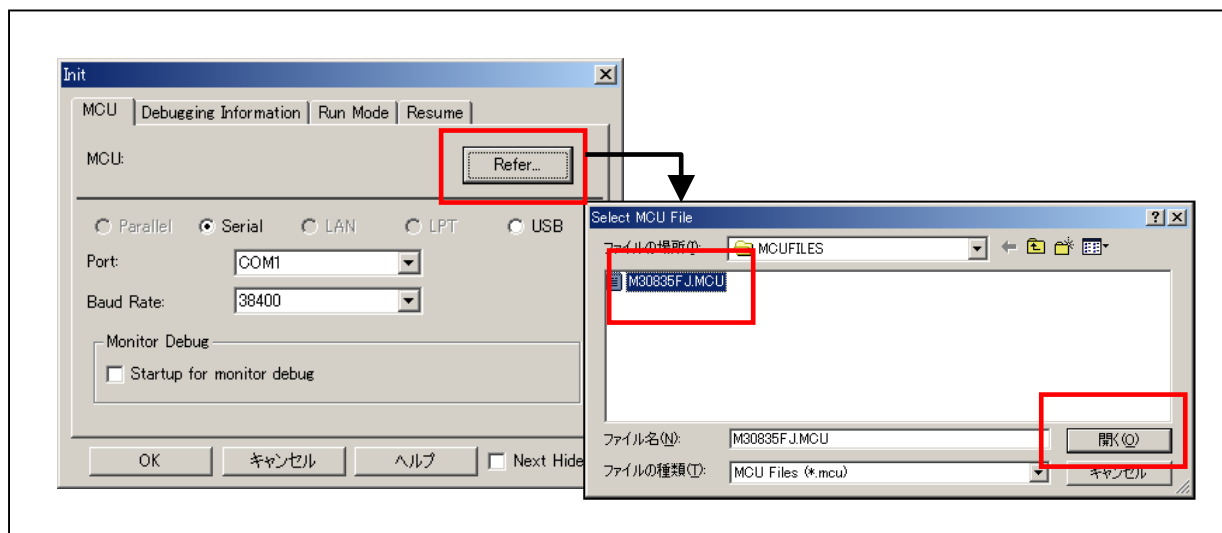


図 2-5 Init ウィンドウ (MCUファイルの指定)

ポート、ボーレート設定を確認したら、ボードをリセットしてInit ウィンドウの[OK]ボタンをクリックします。-hostPCのKD3083とボード側のモニタプログラムが通信を開始します。

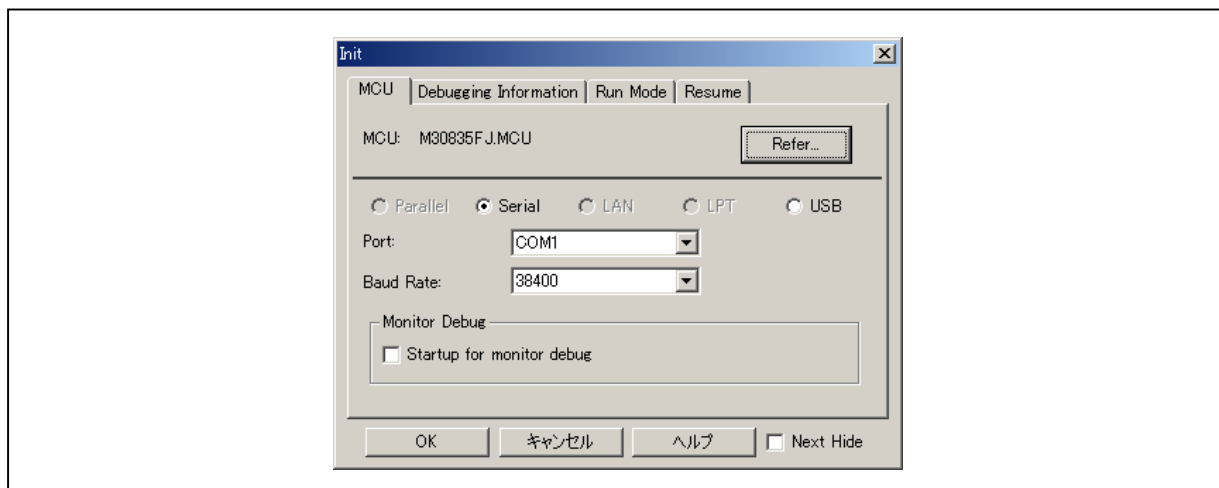


図 2-6 Init ウィンドウ

通信が確立できた場合は、図3-4 のウィンドウがオープンします。[OK]ボタンをクリックして下さい。  
[Processor Mode]では、モニタプログラムがCPUの初期設定を行う際にシングルチップモード、メモリ拡張モード、マイクロプロセッサモードのいずれかに設定するかを選びます。  
ただし、ユーザープログラム実行時はユーザープログラム中でのモード設定となります。  
OAKS32-LANBoardでは、マイクロプロセッサモード (Micro Processor Mode) は選ばないでください。

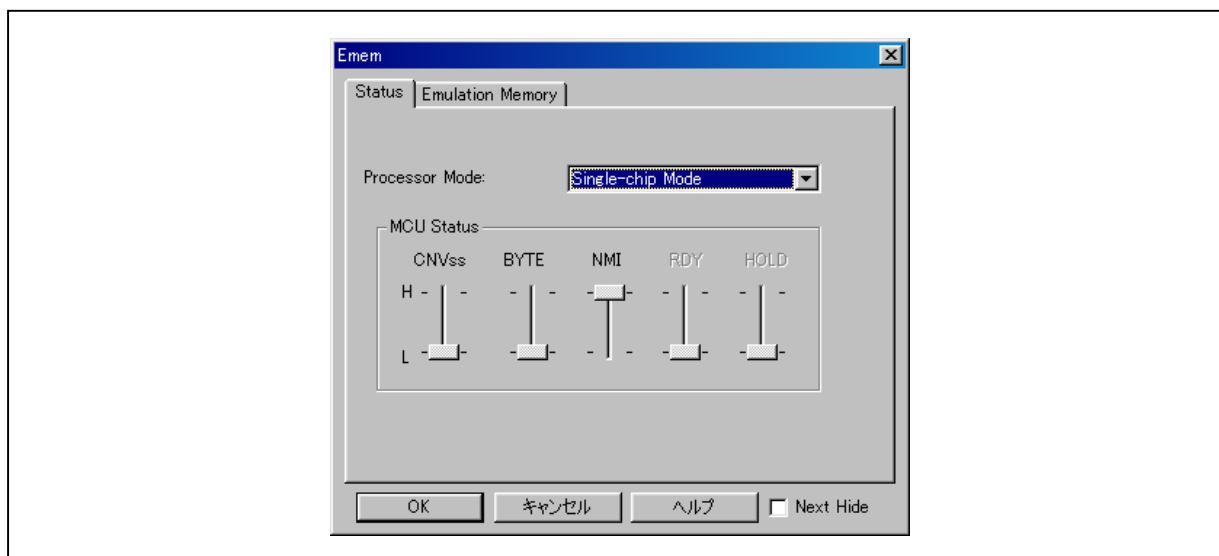


図2-7 KD3083 設定ウィンドウ

モニタプログラムのCPU初期設定後、図2-6 のウィンドウがオープンします。

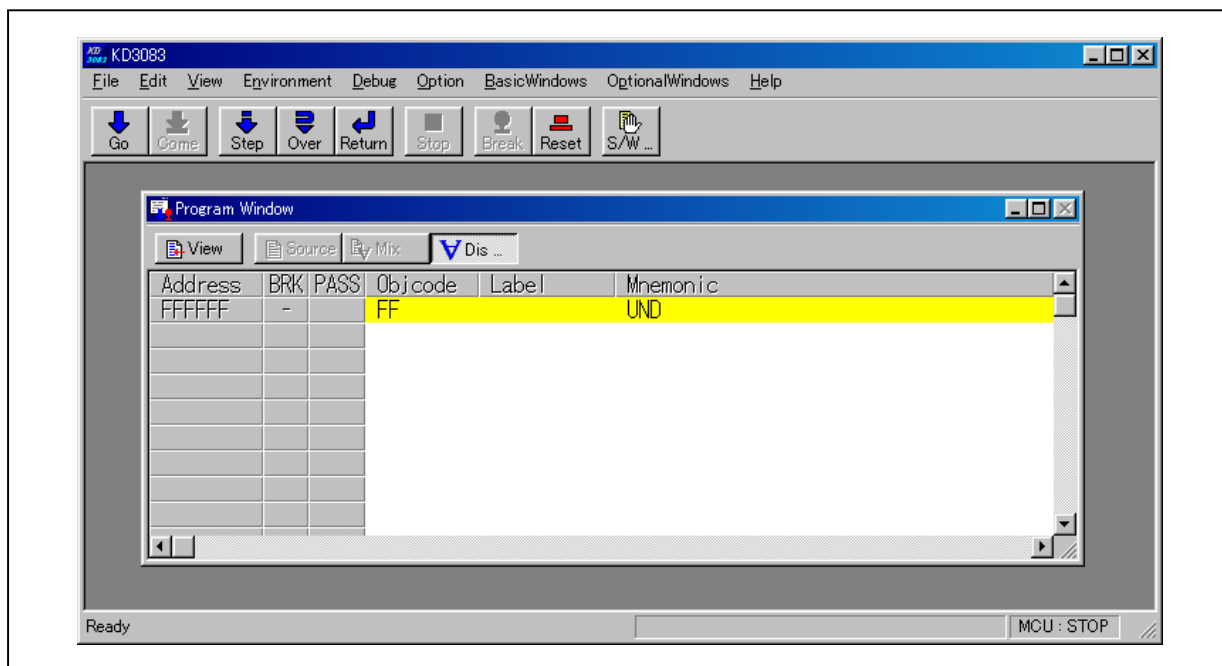


図2-8 KD3083 起動ウィンドウ例

正常に通信できなかった場合は、図2-7 のエラーウィンドウを表示します。



図2-9 起動時のエラーウィンドウ

## 2.2.4 KD3083 の終了

KD3083 を終了するには、[File]—[Exit]メニューを選択します。Exit メニューを選択すると、下図に示すウィンドウがオープンしますので、[OK]ボタンをクリックしてください。[キャンセル]ボタンをクリックすると KD3083 は終了しません。

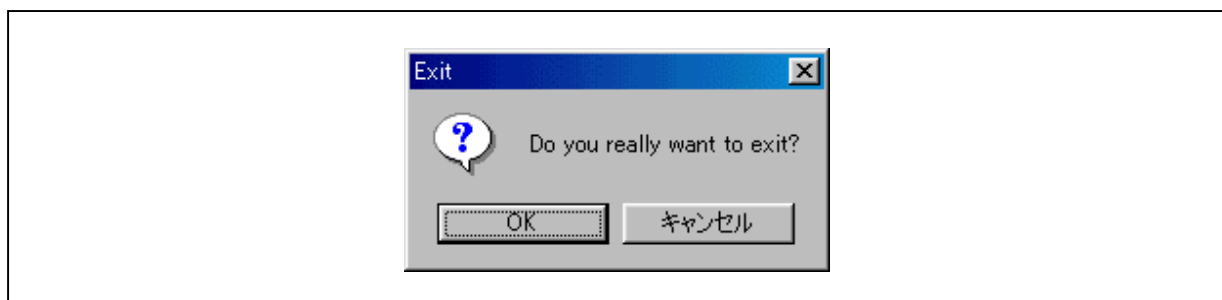


図 2-10 終了ウィンドウ

## 2.2.5 KD3083 のアンインストール

KD3083をアンインストールするには [スタート]—[設定]— [コントロールパネル] を開き、 [アプリケーションの追加、削除] をクリックします。プログラムリストから [KD3083 V3.00 Release 1] を選択し、 [追加と削除] ボタンをクリックします。アンインストールウィンドウが開き、KD3083 がアンインストールされます。

## 2.2.6 KD3083 のエラー対策

### ①起動時に通信エラーが発生した場合の対処

起動時に通信エラーが発生した場合は、以下の内容を確認してください。

- 正しくハードウェアのセットアップが行われているか。
- ケーブルの接続不良がないか、電源が投入されているか。
- Init ダイアログに設定したシリアルポートとキットが接続されているホストパソコンのシリアルポートが一致しているか。

### ②デバッグ中に通信エラーが発生した場合の対処

デバッグ中に通信エラーが発生する場合は、ユーザープログラムの影響（暴走やモニタ領域へのアクセスなど）により、モニタプログラムが暴走したと考えられます。この場合、以下の手順にしたがってシステムを初期化してください。



図2-11 通信エラー 1

1. エラーダイアログを [OK] ボタンを押して閉じる。
2. Exit ダイアログが開いたら [キャンセル] ボタンを押して閉じる。
3. **KD3083** のリセットボタンを押す。

これで、デバッグを再開することができます。ただし、ユーザープログラム領域のデータが破壊されている可能性もありますので、再度プログラムをダウンロードすることをお勧めします。

デバッグ中に通信エラーが多く発生する場合は以下の点を確認してください。

- ・割り込みプログラムを使用しており、多重割り込みを禁止していないか。
- ・割り込みプログラムの処理時間が長く、割り込みプログラムの先頭でI フラグを '1' にしていないか。

### ③ダウンロード中に異常が発生した場合の対処

プログラムのダウンロード中にダウンロードが停止する等の異常が発生する場合、ユーザープログラムの一部（割り込みプログラム等）が動作している可能性があります。

プログラムのダウンロードもしくはリロードをする場合は必ずダウンロード実行前にKD3083上のリセットボタンを押してください。その後、ダウンロードもしくはリロードを実行するようにしてください。

④KD3083 を一度終了したあと、KD3083 を再開しても次のメッセージが出る場合

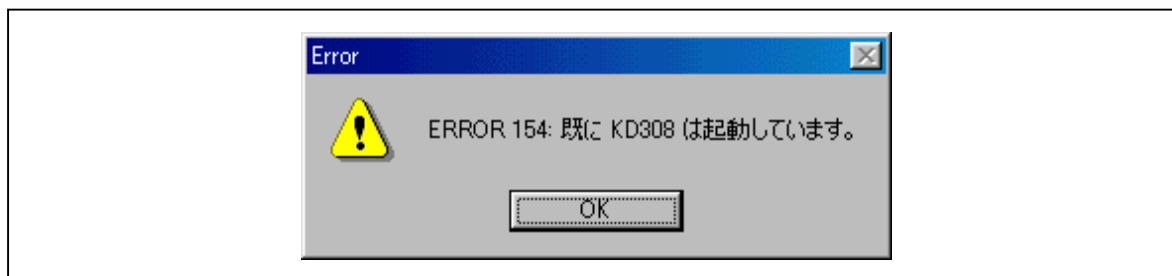


図2-12 通信エラー 2

この場合は、時間を置いて再起動するか、プログラムの強制終了[Ctrl+Alt+Delete]で KD308を終了して下さい。

### 2.2.7 ソースファイルを使用したデバッグを行うために

KD3083 にてシンボルを使用したデバッグを行うためには、NC308WAでコンパイル時に、デバッグオプション `-g` をつけてください。

## 2.3 内蔵フラッシュメモリ書き込みプログラム (Flash Starter) のセットアップ

### 2.3.1 Flash Starter について

CPU内部の書き換えブートプログラムと通信を行い、CPU内蔵のフラッシュメモリを書き換えるためのソフトウェアです。

扱う機械語ファイルはモトローラS2形式 (.mot) のファイルです。

書き換えブートプログラムは内蔵フラッシュメモリとは違う領域に、CPU出荷時に書き込まれています。

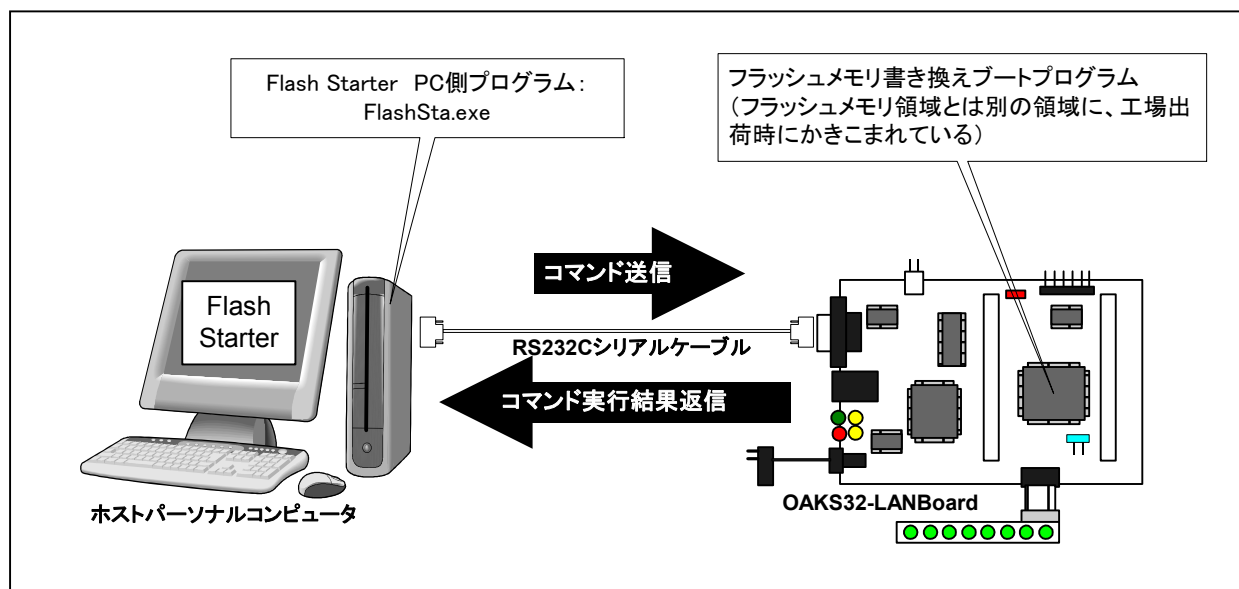


図2-13 Flash Starterのしくみ

OAKS32-LanBoardでは、モニタプログラムとTCP/IPプロトコルスタック (TCP308)とリアルタイムOS (MR308)が含まれた機械語ファイル (stcp308\_mr308mon.mot) を提供しています。この機械語ファイルはOAKS32-LanBoardのCPU内蔵フラッシュメモリにあらかじめ書き込まれていますが、再度書き換えが必要な場合は、Flash Starterを使用して書き換えます。(本章 2.3.3 機械語ファイルの書き込み方参照)

### 2.3.2 Flash Starter のインストール

- ① 任意のディレクトリを作成します。
- ② CD-ROM にある ¥TOOL¥FlashStart ディレクトリ内すべてのファイルを作成したディレクトリにコピーします。

### 2.3.3 Flash Starter の起動

CPU内部の書き換えブートプログラムを起動させるため、Flash Starterを起動する前に、ボード上のJP1コネクタをショートしてから電源を投入してください。



図2-14 JP1コネクタ：ショート

起動はコピーしたファイルの内の「Flashsta.exe」をダブルクリックします。Flash Starterが起動すると、環境設定画面が開きます。使用するポートを設定します。プログラムは「Internal flash memroy」を選んでください。

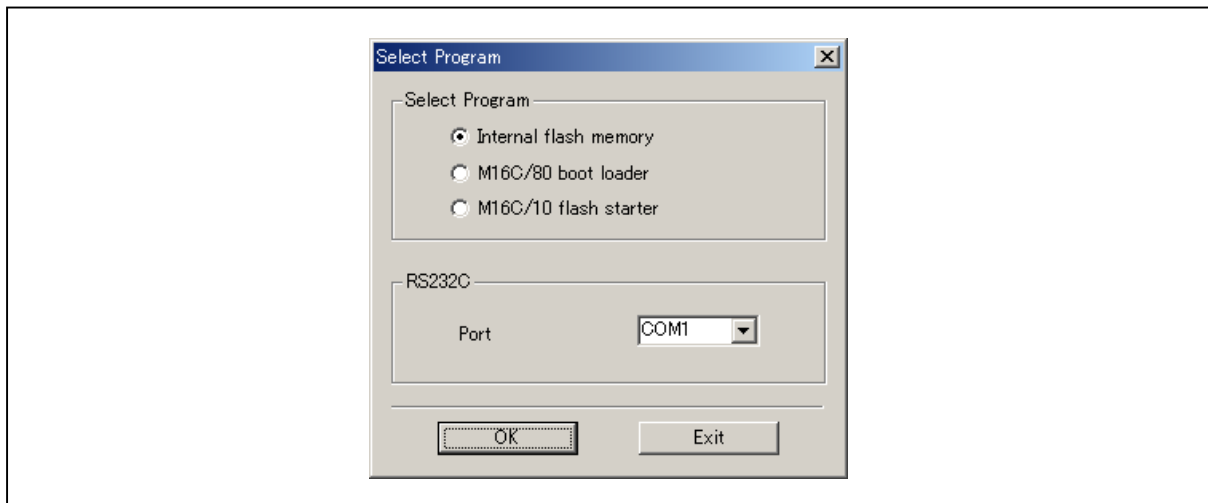


図2-15 Select Programウィンドウ

書き換えブートプログラムとホストPC上の「Flashsta.exe」が通信できない場合は接続エラーウィンドウが開きます。その場合はJP1がショートされているか、ケーブルが正しく接続されているか、電源が投入されているかなど、確認してください。



図2-16 接続エラーウィンドウ

Select Programダイアログの「OK」ボタンをクリックすると、ID Checkダイアログが開きます。

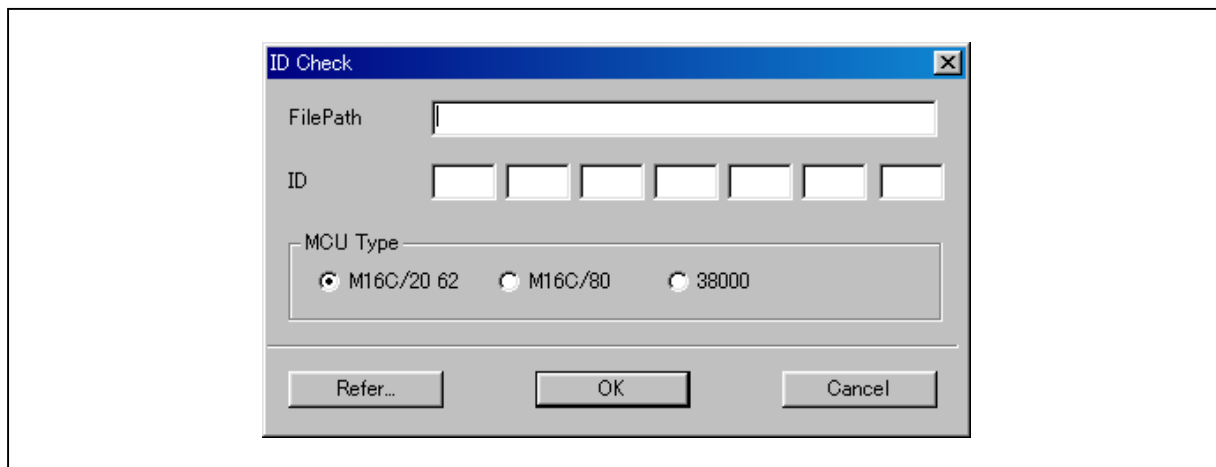


図2-17 ID Checkウィンドウ

### 2.3.4 KD3083 モニタプログラム機械語ファイルの書き込み方

「ID Check」ダイアログの「Refer」ボタンを押してstcp328\_mr308mon.motを選びます。この時、同じディレクトリのstcp328\_mr308mon.idに書き込まれているID番号が自動的に設定されます。

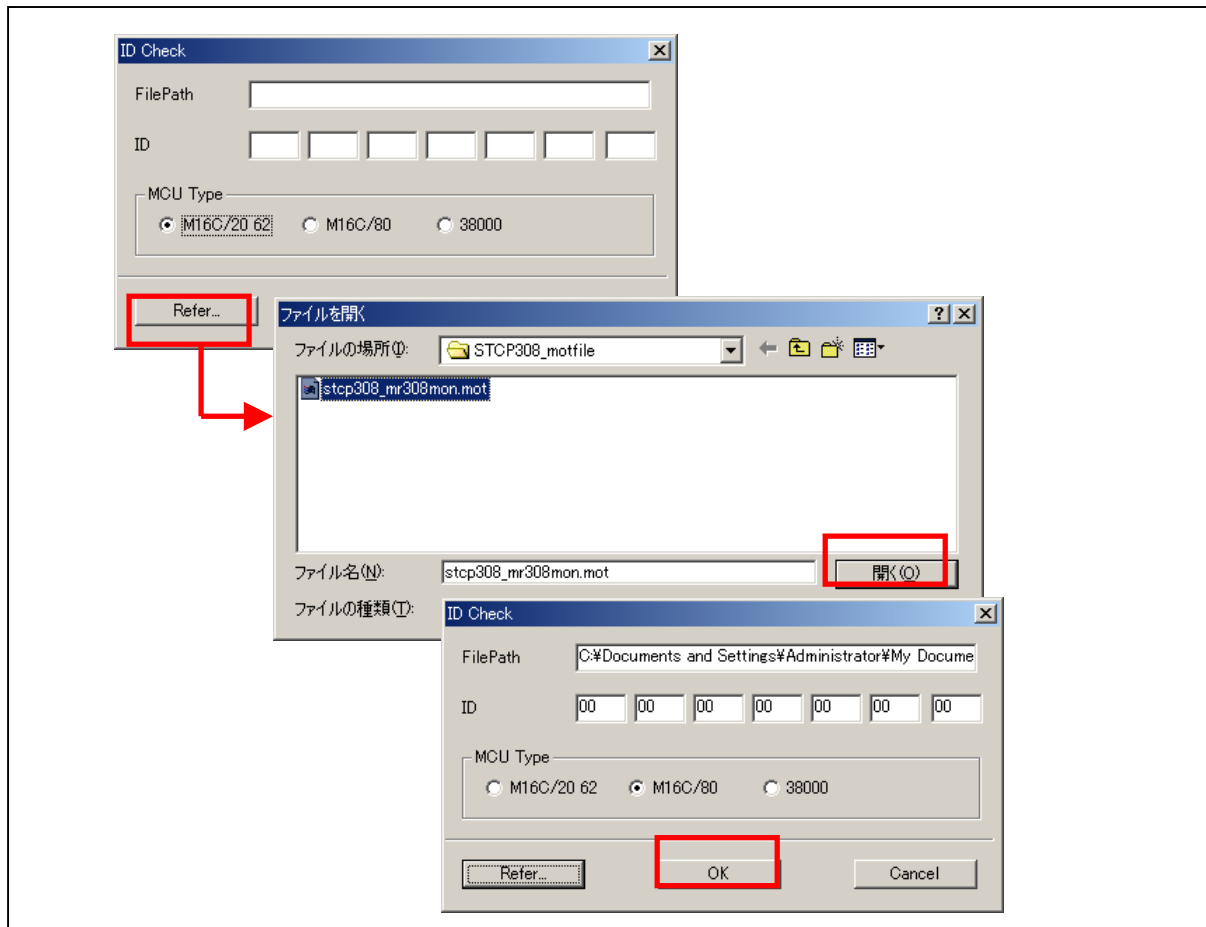


図2-18 機械語ファイルの選択

「デバイスコマンドダイアログ」が開きます。  
「Erase」コマンドでフラッシュメモリのデータを消去します。

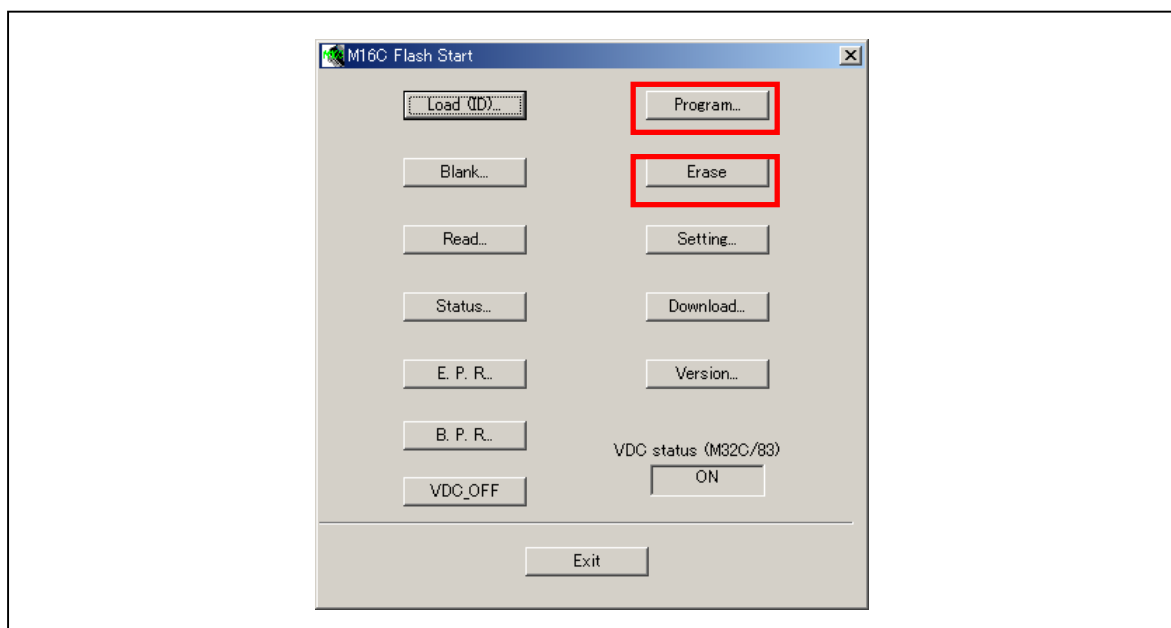


図2-19 デバイスコマンドダイアログ

「Erase?」メッセージが2回出ます。「OK」を押します。

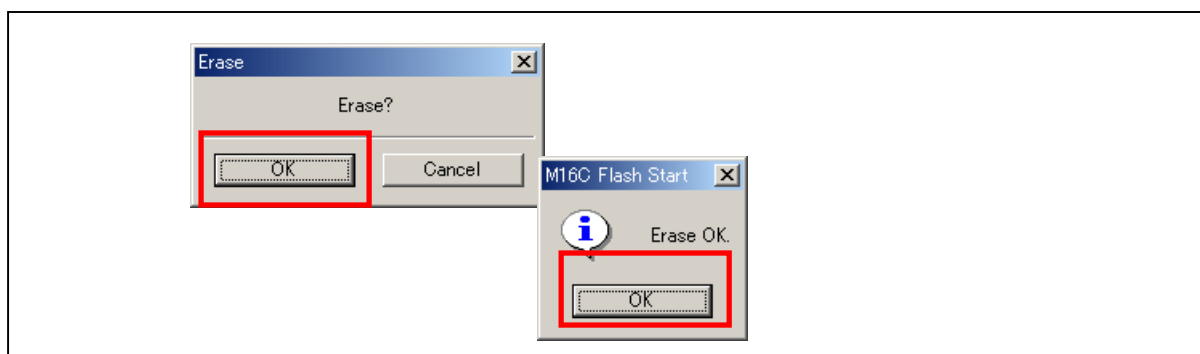


図2-20 Erase

デバイスダイアログの「Program」を押して、選択した機械語ファイルを書き込みます。各確認メッセージの「OK」を押します。

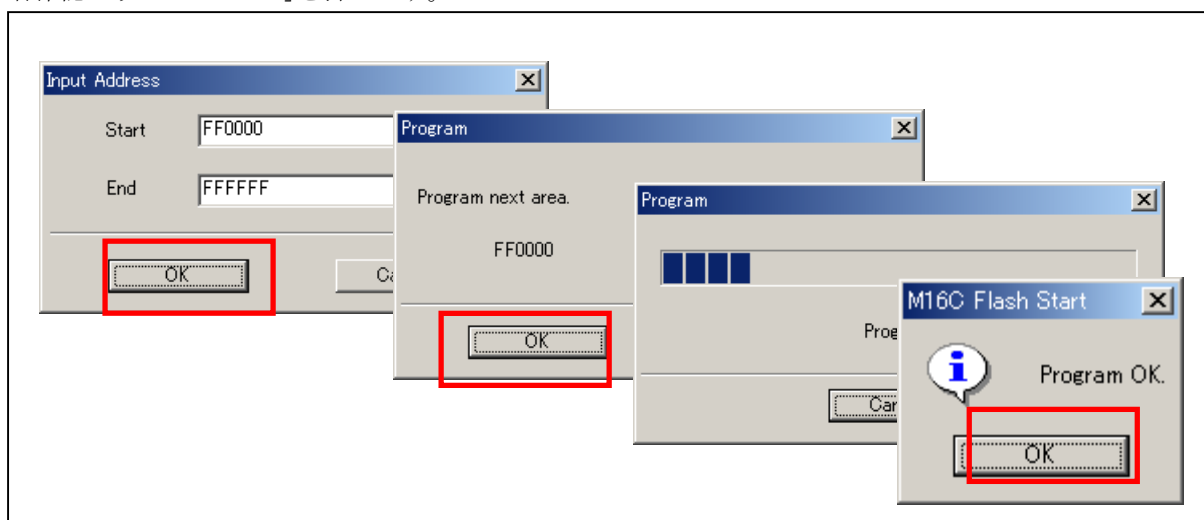


図2-21 デバイスコマンドダイアログ

「デバイスコマンドダイアログ」の「Exit」を押して終了します。  
以上で機械語ファイルの書き込みが終わりました。

KD3083を起動してモニタプログラムと通信できるか試してみます。

ボードの電源をはずし、JP1コネクタをオープンにして再度電源を投入します。KD3083を起動して「Init」ウインドウ → 「設定」ウインドウ → 「起動」ウインドウが開けば正常に書き込まれたことになります。

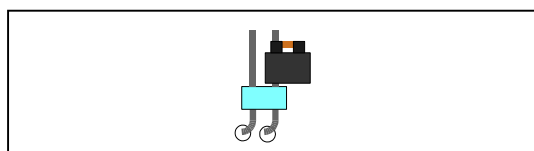


図2-22 JP1コネクタ：オープン

### 2.3.5 Flash Starter のアンインストール

コピーしたディレクトリ以下をすべて消去します。

## 2.4 TM (Tool Manager) のセットアップ

### 2.4.1 TM について

TM (統合化開発環境 : オークス版) は、コンパイラ/デバッガ/エディタなどのツール群を共通グラフィカルユーザーインターフェース (GUI) に統合して、ソフトウェアの開発をスムーズに行うためのツールです。

正規版 (ルネサス テクノロジ株式会社製) に対し、一部機能制限があります。主な制限としては、インスペクタ機能、ライブラリプロジェクト作成機能が使用できません。機能制限の詳細については本キットのCD-ROMに収められているリリースノート (Tools/Rnote/rnote\_tm.pdf) を参照して下さい。

### 2.4.2 TM のインストール

CD-ROM にある ¥TOOL¥TM¥setupディレクトリ内のsetup.exeを実行します。  
インストール画面の指示にしたがってインストールします。

### 2.4.3 TM の起動

Windowsのスタートメニューから「プログラム」-「RENESAS-TOOLS」-「TM V. 3. 20A」-「TM」をクリックします。

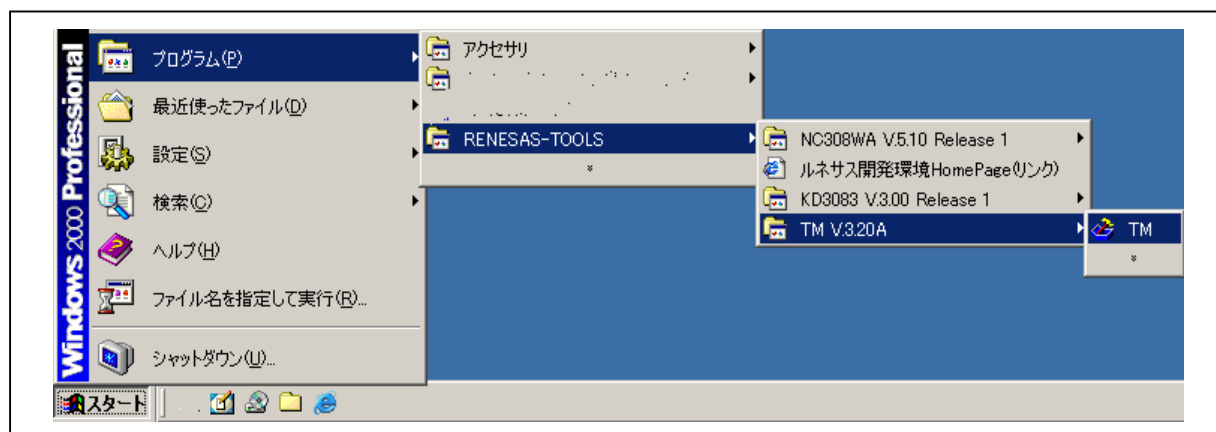


図2-23 TMの起動

プロジェクトバーが表示されます。



図2-24 TM : プロジェクトバー

## 2.4.4 TM のカスタマイズ

OAKS32-LANBoardの開発のためにTMのプロジェクトバーをカスタマイズしましょう。

### ①使用できないボタンを表示しないようにする

TM（オークス版）では、インスペクタ、スタックビューワ、マップビューワ機能が使用できません。プロジェクトバーのこれらのボタンを表示しないように設定します。

プロジェクトバーの「カスタマイズ」ボタンを押して「Button」設定画面を開きます。以下の項目のチェックを外して「OK」を押します。

「Insp:」「関数定義の表示」「関数参照の表示」「変数定義の表示」「変数参照の表示」  
「インスペクタの詳細設定」「スタックの計算」「マップ情報の表示」

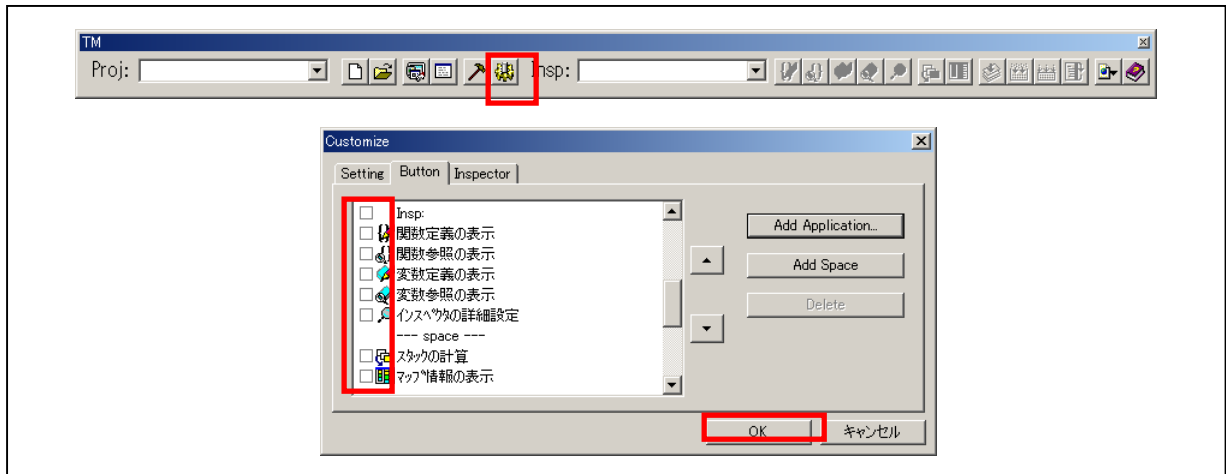


図2-25 TM : ボタンのカスタマイズ

## ②デバッガの登録

デバッガとしてKD3083を登録すると、プロジェクトバーの「デバッガ」ボタンでKD3083を起動することができます。

プロジェクトバーの「ツールの登録」ボタンを押して「Tool Information」ウインドウを開きます。「DUBUG TOOL」ウインドウの「Add」ボタンを押して、「Debug Tool Information」ウインドウの「Browse...」を押します。

インストールディレクトリの¥MTOOLS¥KD3083¥KD3083.exeを選択します。

各ウインドウの「OK」を押して戻ります。

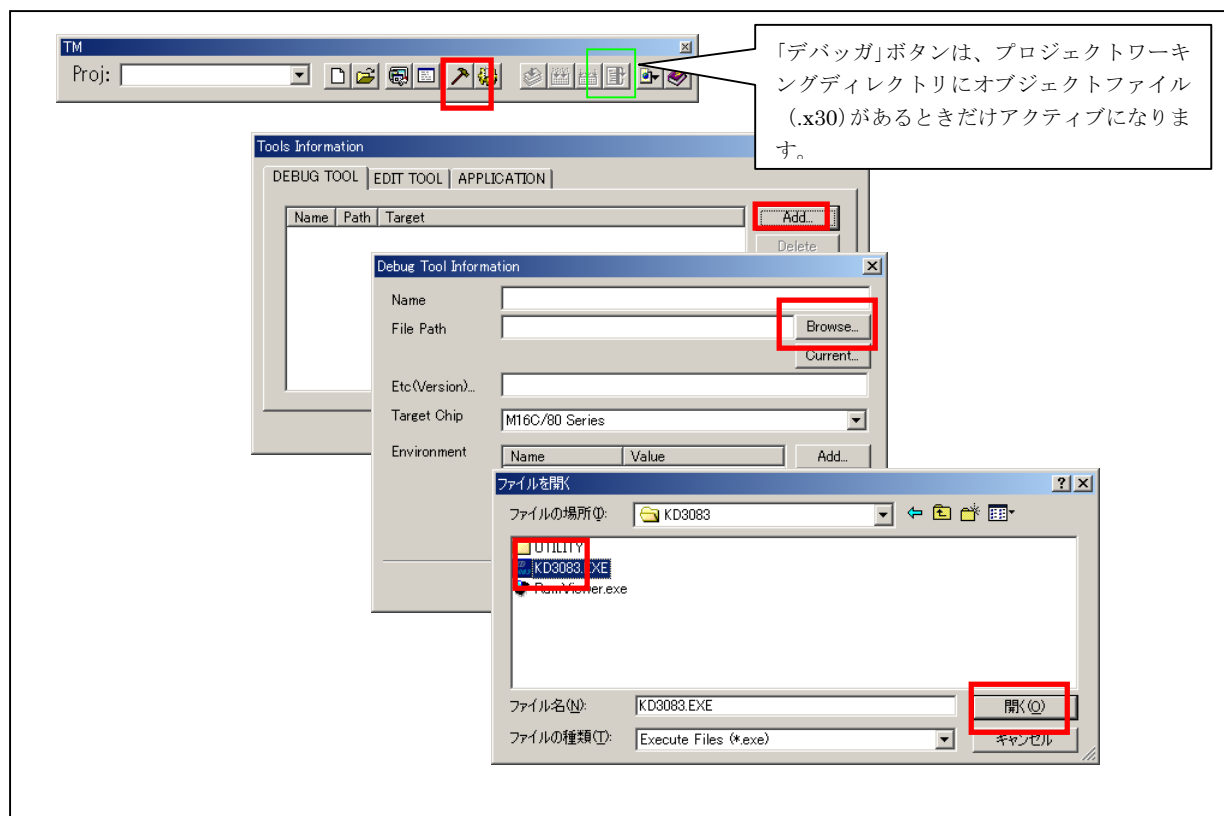


図2-26 TM : デバッガの登録

### ③エディタの登録

普段お使いのエディタを登録するとプロジェクトバーの「エディタを開く」ボタンから起動できます。

プロジェクトバーの「ツールの登録」ボタンをおして「Tool Information」ウインドウを開きます。「EDIT TOOL」ウインドウの「Add」ボタンをおして、「Edit Tool Information」ウインドウの「Browse…」を押します。

ここでは例としてアンカーシステムズ株式会社の「PeggyPro」エディタを登録しています。

「開く」ボタンを押して「Edit Tool Information」ウインドウに戻ります。

「Parameter」の欄にエラータグジャンプのコマンドラインオプションを設定しておく、ビルド実行にエラーが出た場合にエラータグジャンプができます。※1

各ウインドウで「OK」ボタンを押して戻ります。

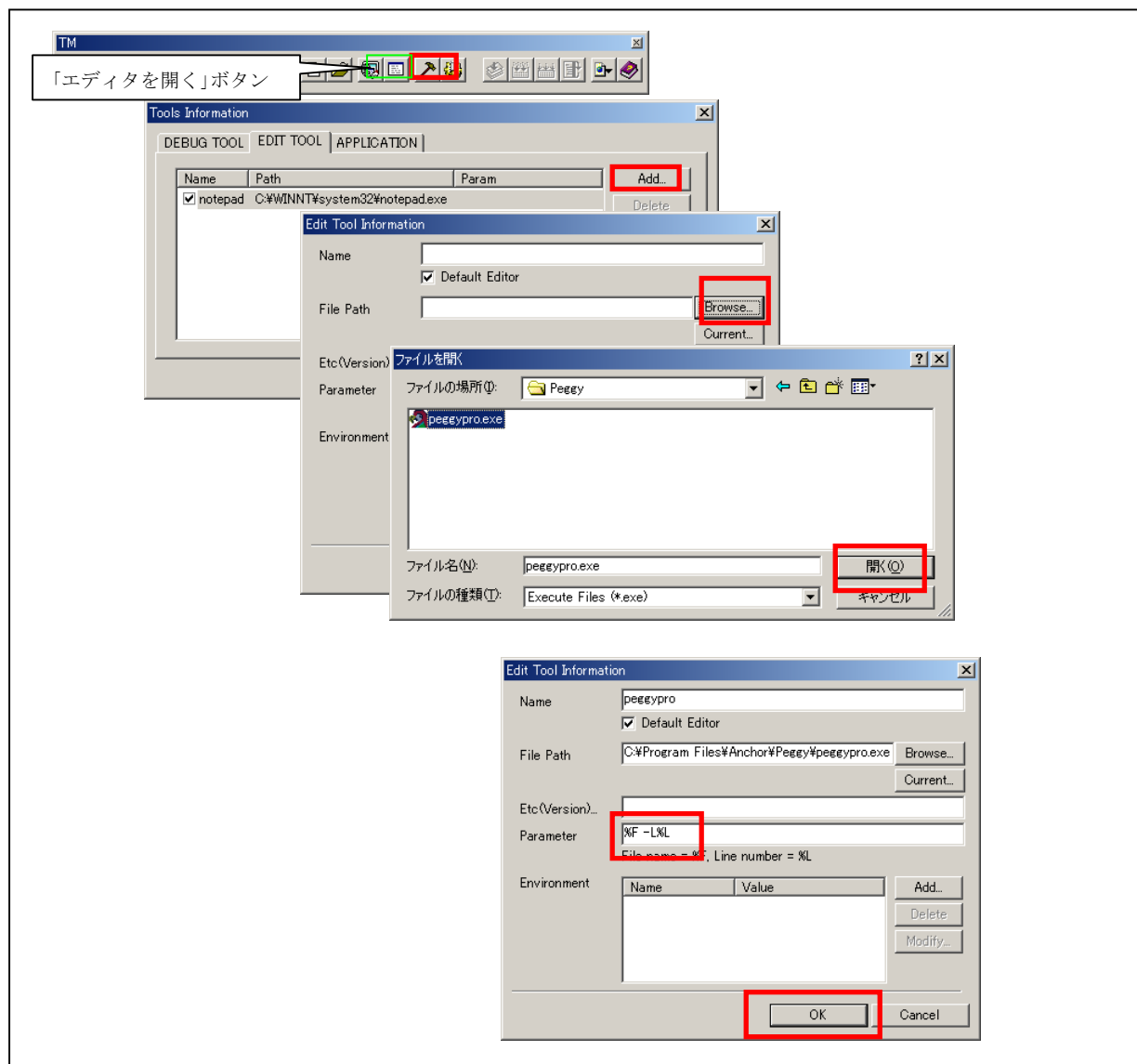


図2-27 TM : エディタの登録

※1 コマンドラインオプションは各エディタによって異なります。

#### ④その他のアプリケーションの登録

「Tool Information」ウインドウの「Application」ウインドウでその他に登録したいアプリケーションを登録することができます。登録した場合、プロジェクトバーに新しいボタンができます。Flash Starterを登録した場合、下図のようになります。

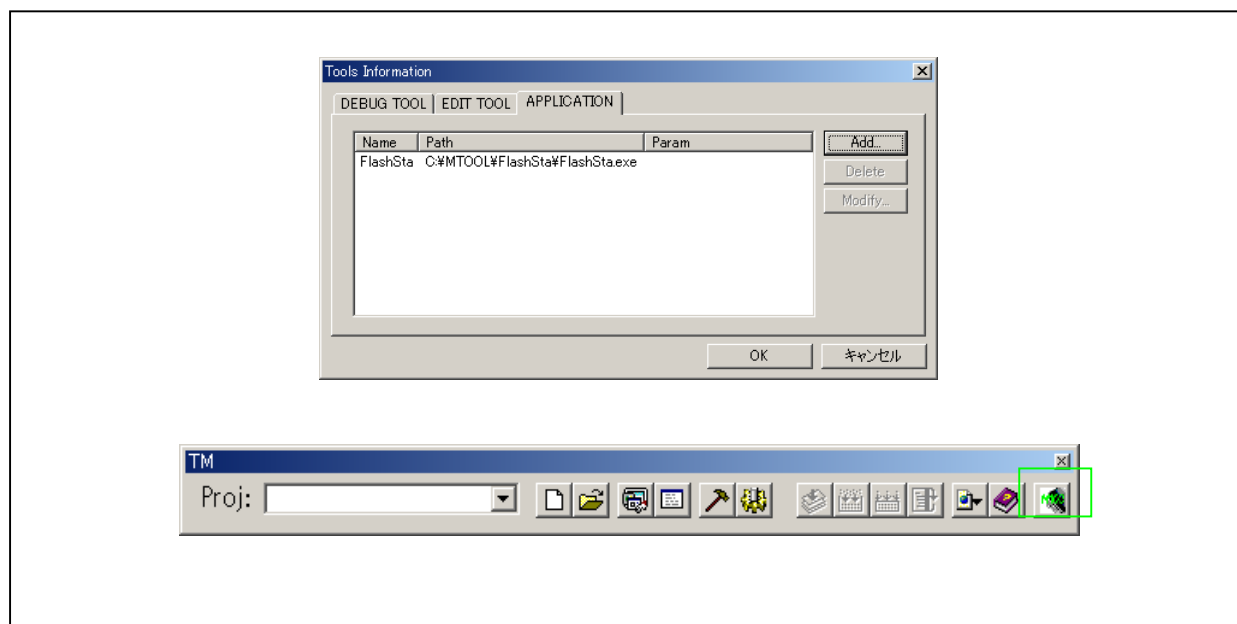


図2-28 TM : Flash Starterの登録

#### 2.4.5 TM のアンインストール

TMをアンインストールするには [スタート]—[設定]— [コントロールパネル] を開き、 [アプリケーションの追加、削除] をクリックします。プログラムリストから [TM V3.20] を選択し、 [追加と削除] ボタンをクリックします。アンインストールウインドウが開き、TMがアンインストールされます。

## 2.5 拡張 LED ボードの組み立て

LED、抵抗、コネクタを基盤にはんだ付けしてください。以下の点に注意してください。

取り付ける順番はとくにありませんが、抵抗 (R0) とコネクタが接している個所があるのでコネクタを先に取り付けたほうがよいかもしれません。

- ・LEDには方向があります。足の長い方 (アノード) を基盤の+表示の方向にとりつけてください。
- ・コネクタにも上下の方向があります。コネクタ側面3箇所の出っ張りがボード本体のCN3コネクタの溝とかみ合います。各出っ張りがCN3コネクタに合う状態で基盤の裏側からとりつけてください。

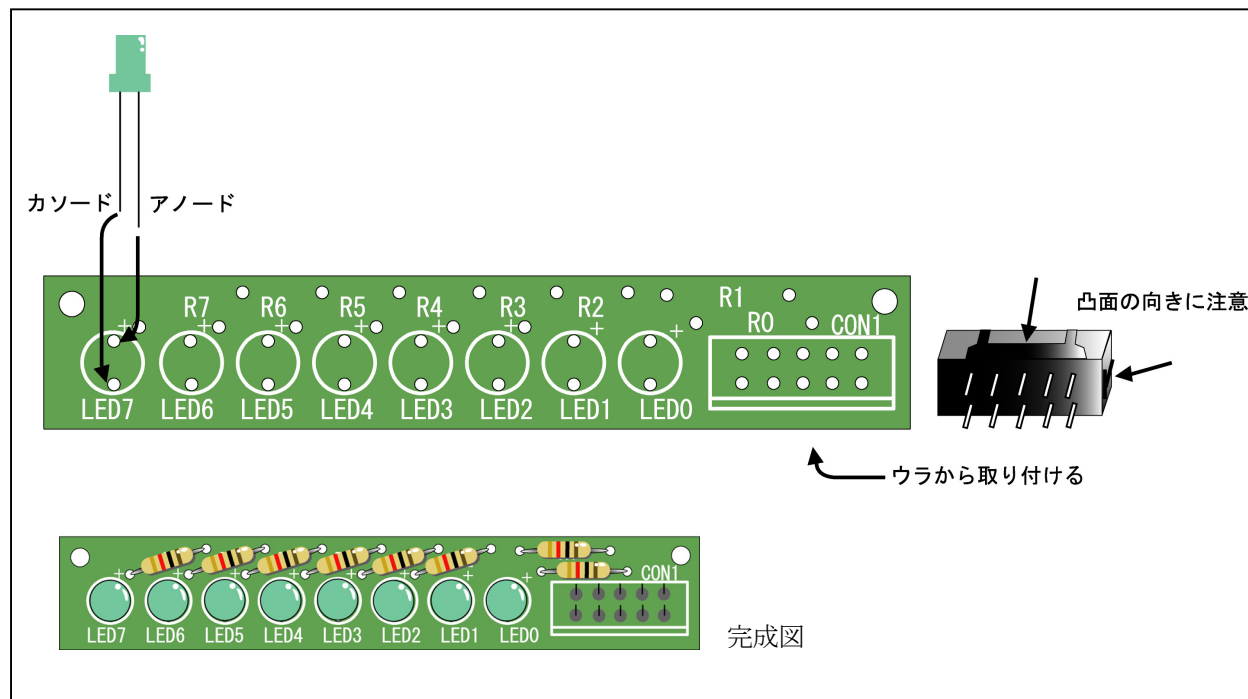


図2-29 拡張LEDボード

ボード本体のコネクタ (CN3) に取り付けてください。

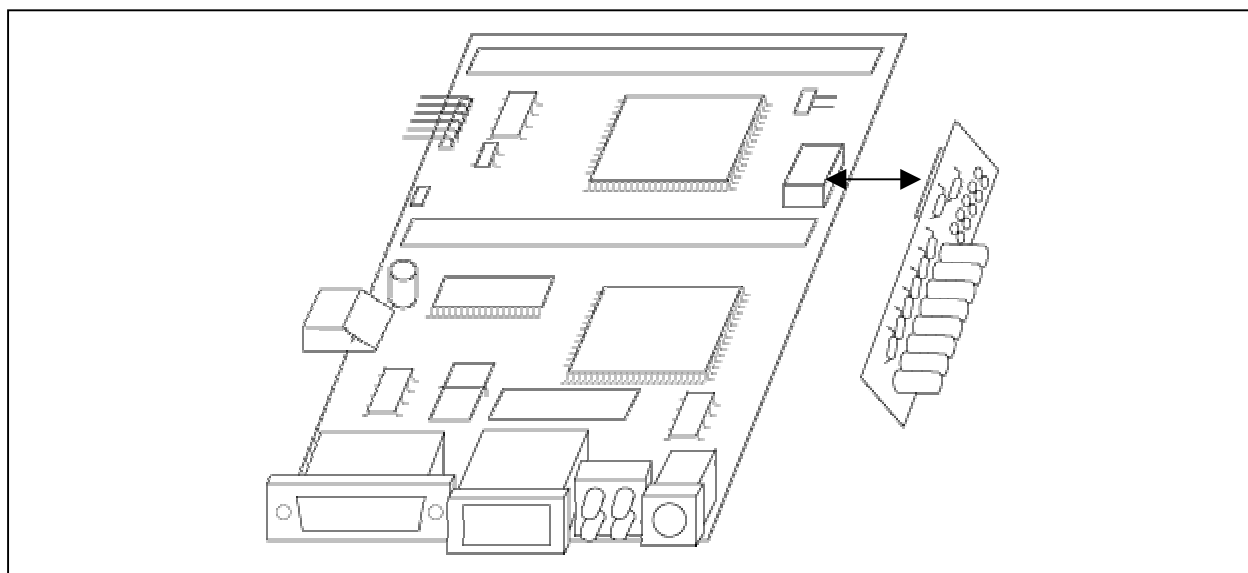


図2-30 拡張LEDボードの本体ボードへの取り付け

拡張ボードをOAKS32-LANBoardに接続すると、LED 0～7は、ポート1の0～7ビットにそれぞれ接続されます。

LEDを点灯させるには、ポート1の該当するビットに“1”を出力させてください。

拡張ボードの動作確認はサンプルプログラムを動作させてチェックしてください。

サンプルプログラムの動作方法については「第3章」を参照してください。

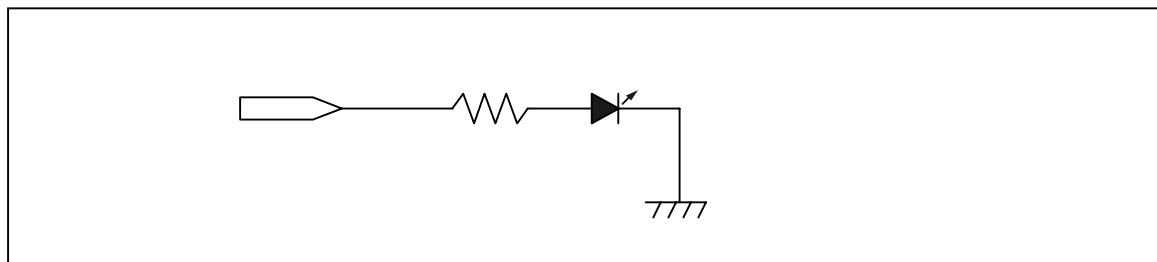


図2-31 LED回路図

## 3. サンプルプログラムを動かしてみよう

### 3.1 サンプルプログラムの内容

開発環境が整ったらサンプルプログラムを動かしてみましょう。付属のCD内のサンプルプログラムをホストPCのハードディスクなど、書き換え可能なドライブにコピーしてください。

OAKS32-LANBoardでは以下のサンプルプログラムを提供しております。ディレクトリの最後に“\_tm”とあるものは、TM上でビルドできるように設定したものです。

表3-1 サンプルプログラム

ディレクトリ (Sampleディレクトリ以下)	内容
oaks32_led_test_tm	拡張ボードのLEDをシフト点灯させます。
oaks32_led_int_tm	oaks32_led_test_tmと同じ動作でタイマ割り込み使用
oaks32_mr308_tm1	oaks32_led_test_tmと同じ動作でMR308使用 (周期起動ハンドラ)
oaks32_mr308_tm2	oaks32_led_test_tmと同じ動作でMR308使用 (イベントフラグ)
oaks32_mr308_tm3	oaks32_led_test_tmと同じ動作でMR308使用 (時間管理機能)
oaks32_stcp308_tm1	STCP308でping、ARPを確認するためのサンプルです。
oaks32_stcp308_tm2	STCP308でTELNETコマンドを実現 STCP308正規版で提供されるサンプルを本キットにインプリメントしたものです。
oaks32_stcp308_make2	上記サンプルをMAKEでコンパイルできるようにしたものです。 MAKEファイルの設定についてはstcp308リリースノート STCP308¥doc¥stcp308subj.pdfを参照してください。
oaks32_stcp308_tm3	oaks32_stcp308_tm2のTELNETコマンドを変更したものです。 TELNETコマンドでLEDを制御します。

サンプルプログラムをCDからハードディスク等にコピーすると全てのファイル、ディレクトリのプロパティは“読み取り専用”となっています。TMでプロジェクトを開く前に必ず全てのファイル、ディレクトリのプロパティの“読み取り専用”チェックをはずしてください。(ファイル、ディレクトリを選んで右クリックするとメニューが出ます。プロパティを選んでください。)

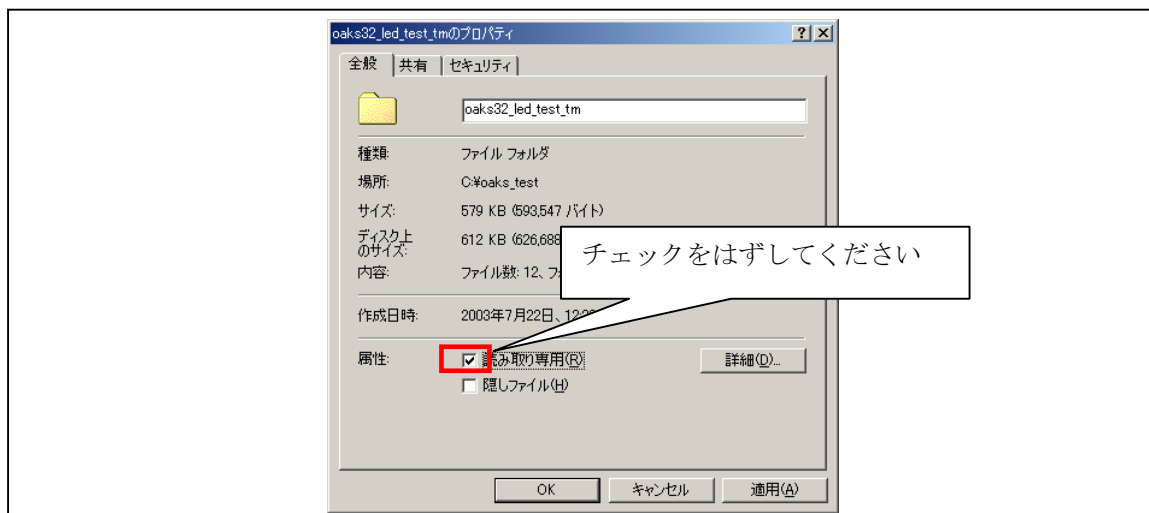


図3-1 ディレクトリ/ファイルのプロパティ

## 3.2 LED テストプログラムを動かす

付属のCD-ROMに収録されているLEDのテストプログラム (SampleYoaks32\_led\_test\_tmフォルダ内) をTMでビルドし、KD8083でダウンロード、動作させてみましょう。

### 3.2.1 LED テストプログラムの動作

100msごとに拡張LEDボードのLED1からLED2、LED3と順番に点灯します。時間計測はソフトウェアによるウエイトでおこなっており正確ではありません。

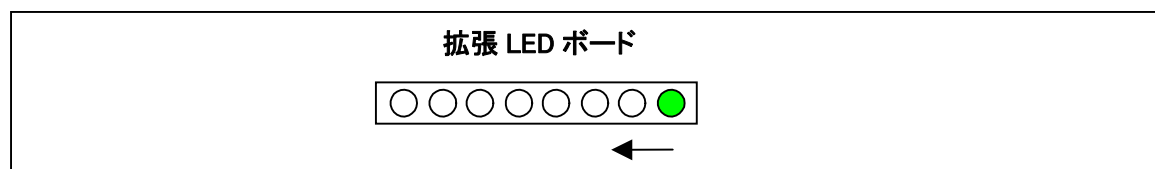


図3-2 LEDサンプルプログラム

### 3.2.2 LED テストプログラムのファイル構成

以下にLEDテストプログラムのファイル構成を示します。

表3-1 LEDサンプルプログラムファイル構成 (oaks32\_led\_test\_tm)

ファイル (ディレクトリ) 名	内容	
ncrt0. a30	初期設定プログラム	
led_test. c	メインのプログラム	
sect308. inc	メモリ空間のセクション定義のファイル (ncrt0. a30内でインクルード)	
sfr_o32. h	SFR (スペシャルファンクションレジスタ) のシンボル定義ファイル (ncrt0. a30内でインクルード)	
Work (ディレクトリ) TMのビルドによって生成されたファイルが格納されます。	ファイル名	内容
	led_test. tmk	TMの設定ファイル
	led_test. tmi	
	led_test. cmd	

### 3.2.3 プログラムのビルドから動作させるまでの手順

#### ①サンプルプログラムをコピーする

oaks32\_led\_test\_tmフォルダごとホストPCのハードディスクにコピーしてください。すべてのディレクトリ、ファイルのプロパティの“読み取り専用”チェックをはずしてください。

#### ②TMを起動、プロジェクトを開く

サンプルプログラムはプロジェクト作成、設定済みになっています。

Windowsのスタートメニューから「プログラム」-「RENESAS-TOOLS」-「TM V. 3. 20A」-「TM」をクリックします。TMが起動したら、プロジェクトバーの「Open Prlject」ボタンをクリックし、プロジェクト設定ファイル(oaks32\_led\_test\_tm\workのled\_test.tmk) を選択し、「開く」ボタンをクリックしてください。

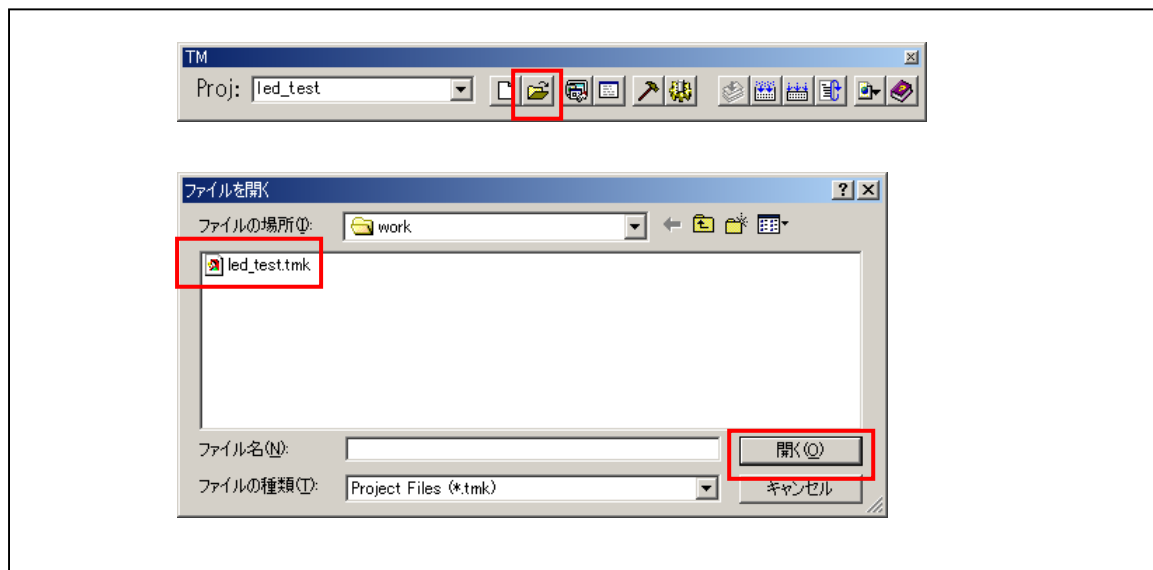


図3-3 TM : プロジェクトを開く

#### ③ビルドする

プロジェクトを開いたら、「ビルド」ボタンをクリックしてください。ビルダが起動しビルドします。終了したらworkディレクトリにled\_test.x30等、ファイルが生成されているのを確認してください。

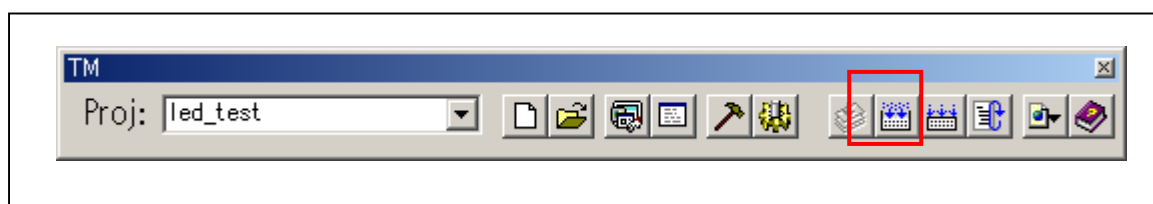


図3-4 TM : ビルド

### ③OAKS32-LANBoardをセットアップする

OAKS32-LanBoardにRS232Cケーブル、拡張LEDボードを接続し、J1コネクタがオープンになっていることを確認してから電源コネクタを接続してください。

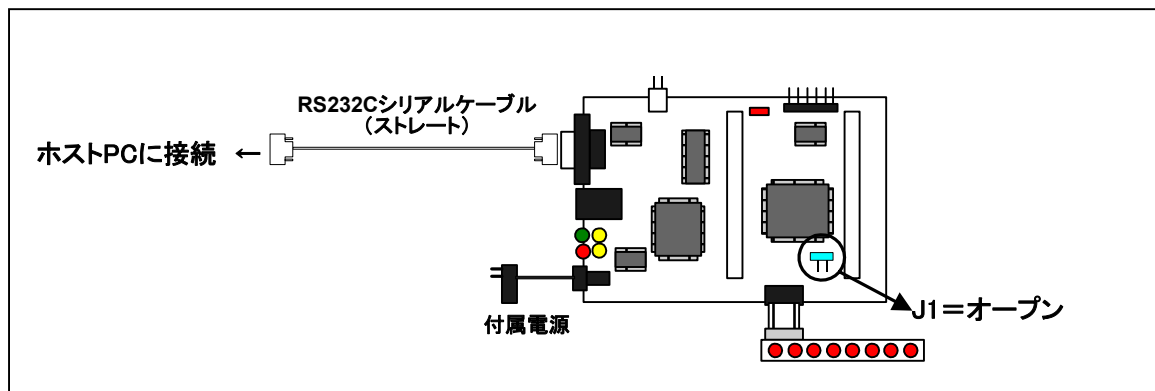


図3-5 OAKS32-LANBoardのセットアップ

### ④KD3083を起動する

TMにKD3083をデバッグツールとして登録済みの場合はTMより起動できます。プロジェクトバーの「デバッガ」ボタンをクリックしてください。

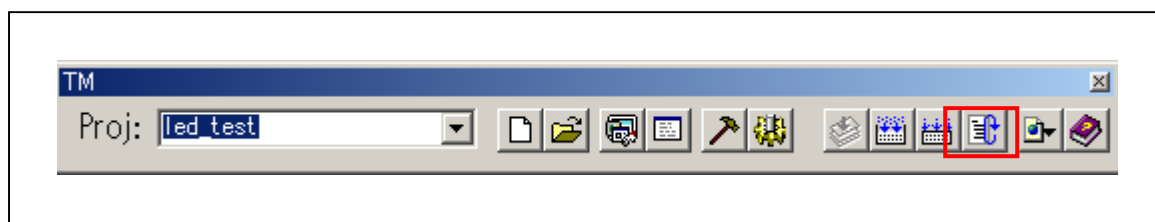


図3-6 TM : TMからKD3083の起動

KD3083が起動しInitダイアログが開きます。この時点ではまだモニタプログラムとは通信していません。MCUプロセッサモードが「Single-chip Mode」になっていることを確認し、ボードのリセットボタンを押してからOKボタンを押してください。モニタプログラムとの通信を開始します。

(MR308、STCP308を使用したプログラムを実行させる場合は「Memory Expansion」に設定してください。)

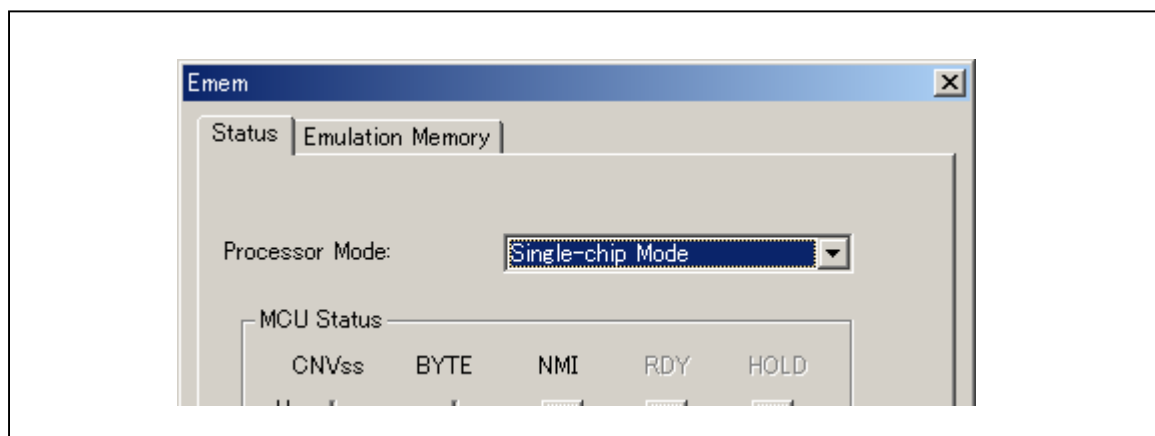


図3-7 KD8083 : Initダイアログ

### ⑤プログラムをダウンロードする

「File」→「Download」→「Load Module」で実行ファイル（sample.x）を選択し、ダウンロードします。

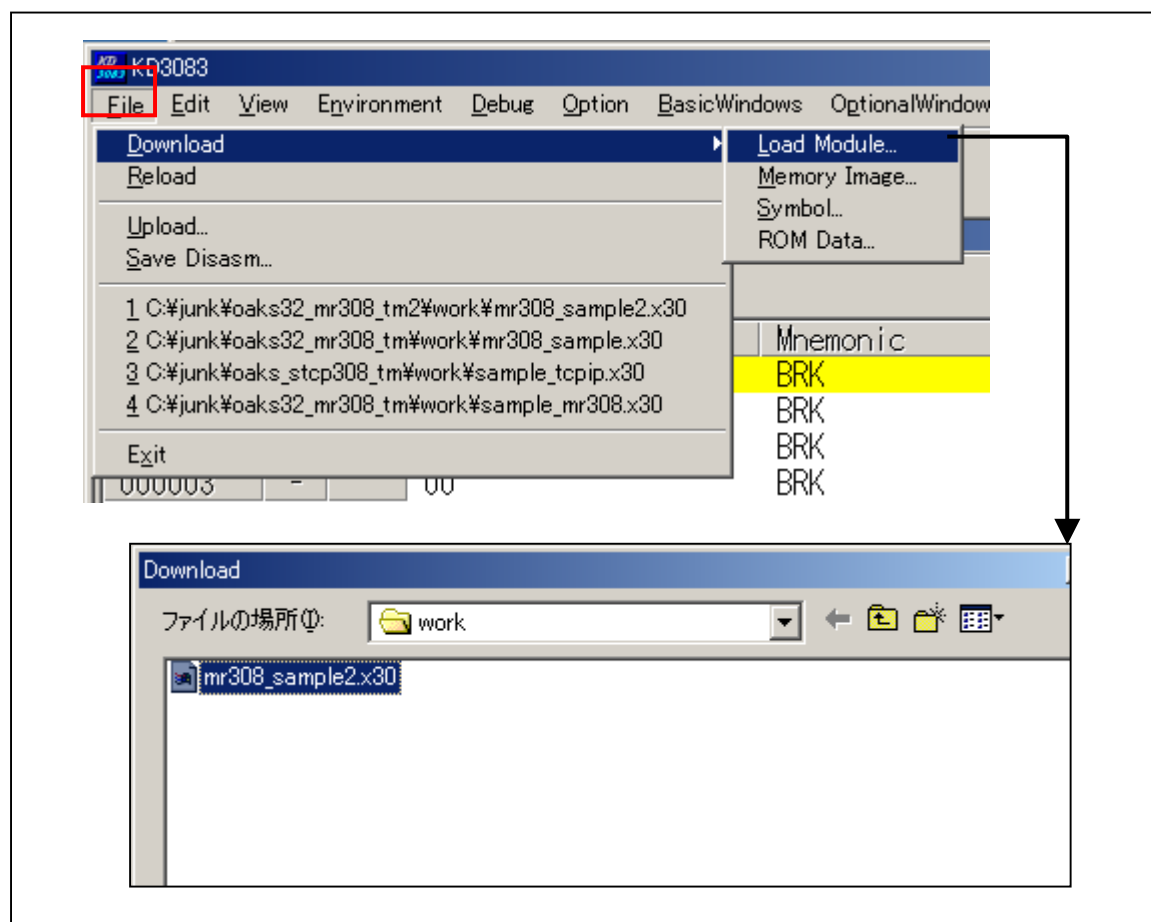


図3-8 KD3083 : プログラムのダウンロード

### ⑥プログラムを実行する

「GO」ボタンをクリックしてプログラムを実行します。

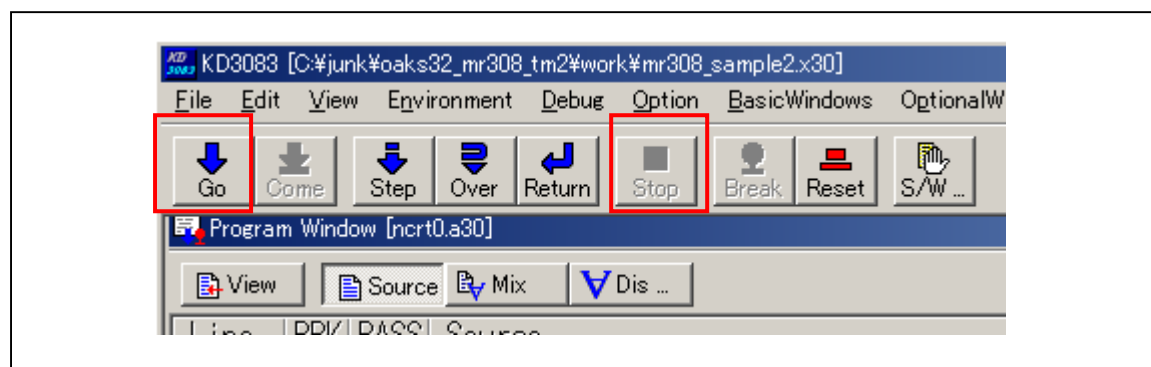


図3-9 KD3083 : プログラムの実行と停止

LED 1～8 がシフトしながら点灯することを確認してください。「STOP」ボタンで停止します。

### ⑦実機でプログラムを動かしてみる

KD3083を終了してリセットし、約5秒間経過するとフラッシュメモリに書き込んだサンプルプログラムが動作します。

KD3083とOAKS32-LANBoardの通信が確立しない場合、以下の事を確認してください。

- 正しくハードウェアのセットアップが行われているか。
- ケーブルの接続不良がないか、電源が投入されているか。
- Init ダイアログに設定したシリアルポートとキットが接続されているホストパソコンのシリアルポートが一致しているか。

製品の出荷には万全を期しておりますが、まれに予め書き込まれているモニタプログラムのデータが壊れている場合があります。上記について確認しても解決しない場合は、モニタプログラムを書き込んでみてください。モニタプログラムの書き込み方については「第2章 2.3.3 機械語ファイルの書き込み方」を参照してください。

## 4 MR308 を使ったプログラム開発

OAKS32-LANBoardでMR308を使用したプログラムを開発する場合の手順、注意事項について説明します。

### 4.1 MR308 の特徴

MR308（オークス版：以下MR308と呼びます）は正規版のMR308（ルネサステクノロジ社製）のサポートするタスクや機能を限定し、MR308カーネルを機械語形式で提供したものです。

正規版の場合は、ユーザプログラムで使用するタスク数、使用する機能の指定、名前の定義などを記述したコンフィグレーションファイルを作成し、コンフィグレータを実行してMR308カーネルと整合をとるためのファイルを生成させます。ユーザプログラムのコンパイラ、アセンブラ時にこれらのファイルを組み込みます。

MR308では、コンフィグレータによる設定済みとし、MR308カーネルと整合をとるために必要なファイルのみを提供しています。（コンフィグレータは提供されていません）

以下にMR308のコンフィグレーションの内容を示します（サンプルプログラムで使用している機能のみ抜粋しています）。その他の設定については、「MR308 オークス版 リリースノート」（CD-ROM¥MR308¥doc¥mr308subj. pdf）を参照してください。

#### <MR308で使用できる機能とその数>

表4-1 使用できる機能と数

タスク数	16
メールボックス数	8
セマフォ数	8
イベントフラグ数	8
固定長メモリプール数	4
周期起動ハンドラ数	8

#### <システムクロック定義>

表4-2 システムクロック定義

mpu_clock	30 MHz
timer	A0 システムタイマはA0を使用します。
IPL	4 システムクロック割り込み優先レベルです。
unit_time	10ms 時間管理機能など設定する際の基本単位となります。
initial_time	0:0:0

### <タスク定義>

表4-3 タスク定義

entry_address	task1(), task2(), task3(), task4(), task5(), task6(), task7(), task8()
stack_size	300
priority	6
initial_start	task1()のみONで、その他のタスクはOFF ONの場合、システム起動時にready状態となり実行されます。 OFFの場合、システム起動時にはwait状態です。

### <イベントフラグ>

表4-4 イベントフラグ定義

entry_address	flg1, flg2, flg3, flg4, flg5, flg6, flg7, flg8
---------------	------------------------------------------------

### <周期起動ハンドラ>

表4-5周期起動ハンドラ

entry_address	cyc1, cyc2, cyc3, cyc4, cyc5, cyc6, cyc7, cyc8
interval_counter	提供するファイルmr308ent.c中の配列_D_CHCB_CNTに設定します。
mode	TCY_OFF 初期状態は動作していません。act_cycシステムコールを発行することで動作を始めます。

### <セマフォ>

表4-6周期起動ハンドラ

entry_address	sem1, sem2, sem3, sem4, sem5, sem6, sem7, sem8
initial_count	初期値は0 0以外の値を設定したい場合はsig_sem (セマフォを一つ返却する：つまりセマフォカウンタに+1することになります) を設定したい数値の回数だけコールします。 (ハンドラから設定する場合はisig_semをコールしてください)

## 4.2 MR308 で使用するファイル

MR308を使ったプログラム開発をする場合、付属のCD-ROMに収められている以下のファイルを使用してください。

### <フラッシュメモリに書き込んで使用するファイル>

stcp308\_mr308mon.motはOAKS32-LANBoardのCPU内蔵フラッシュメモリに予め書き込まれています。

表4-7 Tools¥Monitor\_programディレクトリ

ファイル名	内容
stcp308_mr308mon.mot	STCP308ライブラリ+MR308カーネル+KD3083モニタ (STC0308、KD3083と共通)
stcp308_mr308mon.mot	FlashStarter用IDファイル

### <ユーザプログラムに組み込んで使用するファイル>

フラッシュメモリに書き込まれているMR308カーネルプログラムとの整合を取るために必要なファイルです。ユーザプログラムでのタスクの先頭アドレス、周期起動ハンドラの周期設定などはmr308ent.cで行います。その他のファイルは基本的に変更せずにそのまま使用します。とくにファイル内の” Don’ tChange” とコメントのある個所に関しては、変更するとMR308が正常に動作しない場合がありますのでご注意ください。

表4-8 MR308¥mr308subディレクトリ

ファイル名	内容
ncrt0.a30	初期設定プログラム (STCP308と共通) 必ずユーザプログラムの一番初めに実行するようにして下さい。
mr308ent.c	MR308エントリアドレス設定ファイル ユーザによる設定が必要なファイルです。
sect308.inc	メモリ空間のセクション設定のファイル (STCP308と共通) ncrt0.a30でインクルードされます。
mr308sym.inc	シンボル定義 ncrt0.a30でインクルードされます。
mr308.inc	アセンブラ言語用インクルードファイル ncrt0.a30でインクルードされます。 ユーザプログラム (アセンブラ言語) でインクルードしてください。
mr308.h	C言語用ヘッダファイル ユーザプログラム (C言語) でインクルードしてください。
Id.h	ID定義 ユーザプログラムでインクルードしてください。
c308mr1m.lib	MR308 C言語インターフェースルーチンライブラリ リンク時にリンクオプションを指定してリンクしてください。

## 4.3 コンパイル時に必要なオプション

コンパイル時に必ず以下のオプションを指定してください。

コンパイラオプション: -Dオプション (パラメータ: far\_msg=1) → -D far\_msg=1

リンクオプション: -Lオプション (指定ファイル: c308mr1m.lib) → -L ..¥c308mr1m.lib

### ※-Dオプション

識別子名を定義します。

### ※-Lオプション

リンク実行時に参照するライブラリファイル名を指定します。指定するファイル名がカレントディレクトリかあるいは環境変数 (LIB30) で指定されているディレクトリ内の場合はファイル名のみを指定します。

## 4.4 TM (Tool Manager) でプロジェクトを作成する手順

TM上でMR308を使用するプログラムを開発する場合の手順を説明します。  
ルートディレクトリCに以下のディレクトリ、ファイルを生成したものとします。

作成したディレクトリ : C:\oaks\sample1\work ※1  
sample1ディレクトリはソースプログラムを格納する  
workディレクトリはTMでビルド後に生成されるファイルを格納する

作成したプログラム : C:\oaks\sample1ディレクトリに” sample1.c” を作成したものとします。

※1プロジェクトのディレクトリ名について  
TMは空白分を含むディレクトリは指定できませんので、ご注意ください。  
例：「My Documents」、「Program Files」等は指定できません

①付属のCD-ROMのMR308\mr308subフォルダより以下のファイルをホストPCにコピーする。

表4-9 MR308を使ったプログラムのファイル構成例

ディレクトリ	ファイル (ディレクトリ)	
C:\oaks\sample1	sample1.c ncrt0.a30 mr308ent.c mr308.inc mr308.h sect308.inc mr308sym.inc id.h mr308.h mr308mrlm.lib	作成したプログラム コピーしたファイル
	work	ビルドで生成されたファイルを格納するディレクトリ

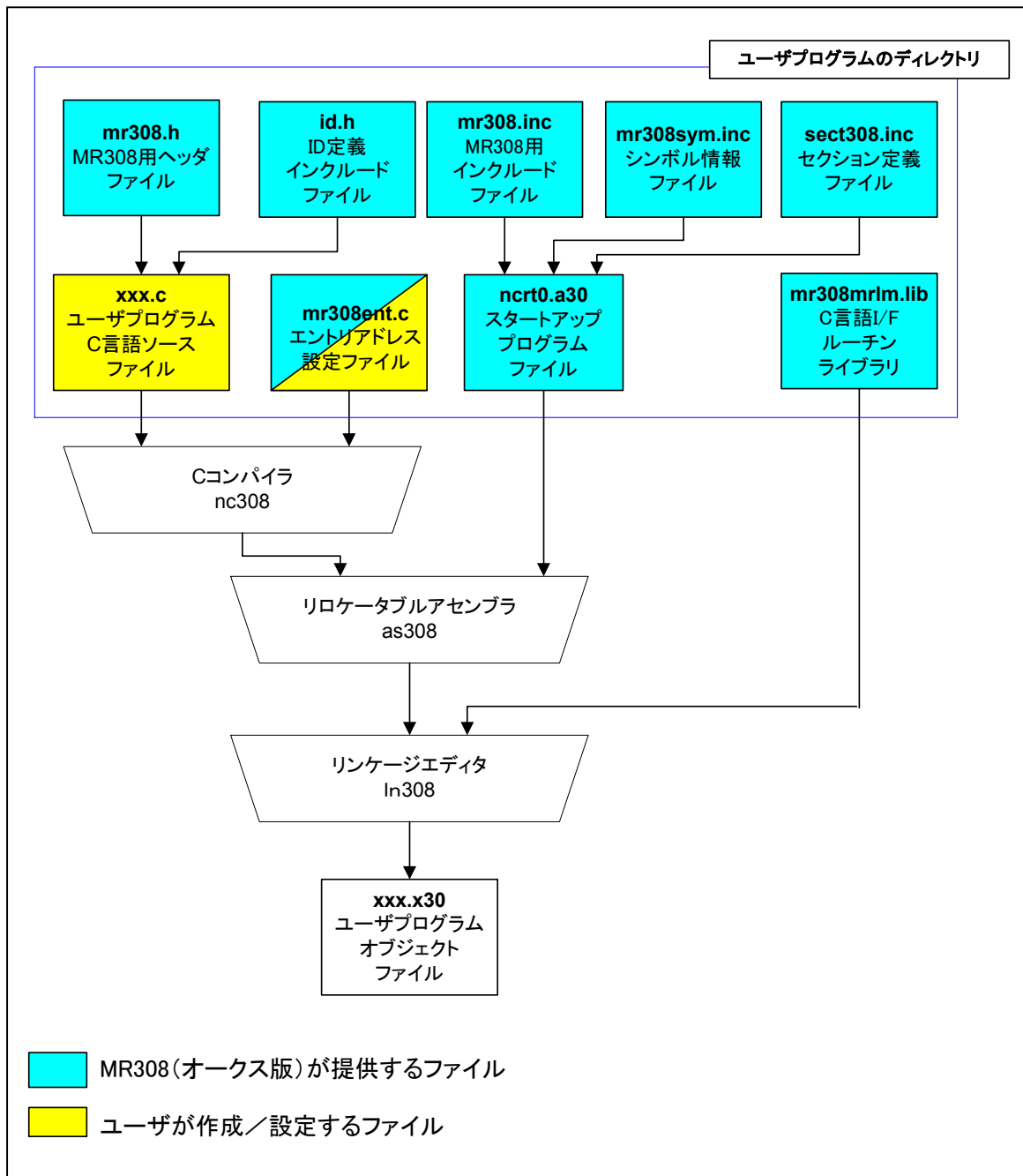


図4-1 MR308オブジェクトファイル生成フロー図 (プログラムをC言語で記述した場合)

## ②TMを立ち上げ、プロジェクトを作成する

「New Project」ボタンをクリックし、ウィザードに従ってプロジェクトを作成します。

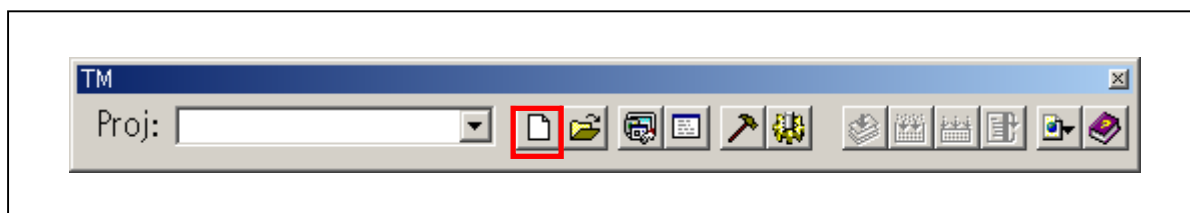


図4-2 TM New Projectボタン

ウィザードのステップ1の設定は以下のように設定してください。

- ・ターゲットチップ：M32C/80Series
- ・プロジェクト名：sample1とします。生成されるオブジェクトファイル名に反映されます。
- ・ワーキングディレクトリ：C:\%oaks%\sample1\workとします。ビルドによって生成されるファイルが格納されます。

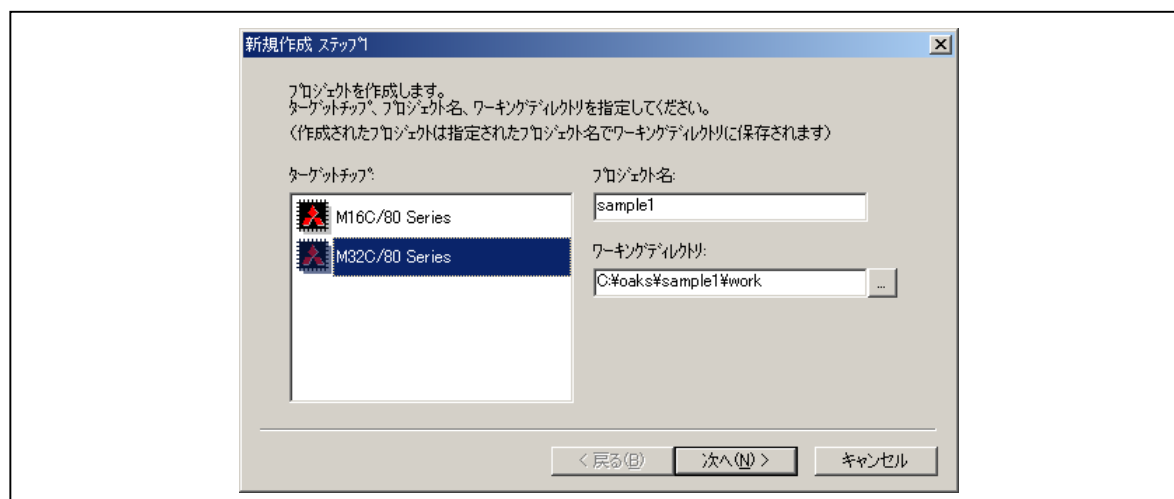


図4-3 TM：新規作成ステップ1

ステップ2の設定は「C言語プロジェクト」を選択してください。

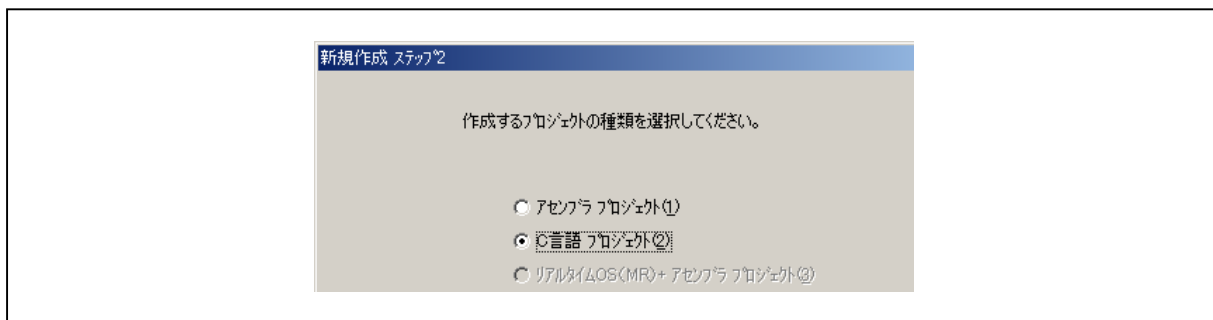


図4-4 TM：新規作成ステップ2

ステップーコンパイラでは、スタートアッププログラムを「カスタム」に指定して、CD-ROMよりコピーした初期設定プログラムを選択してください。

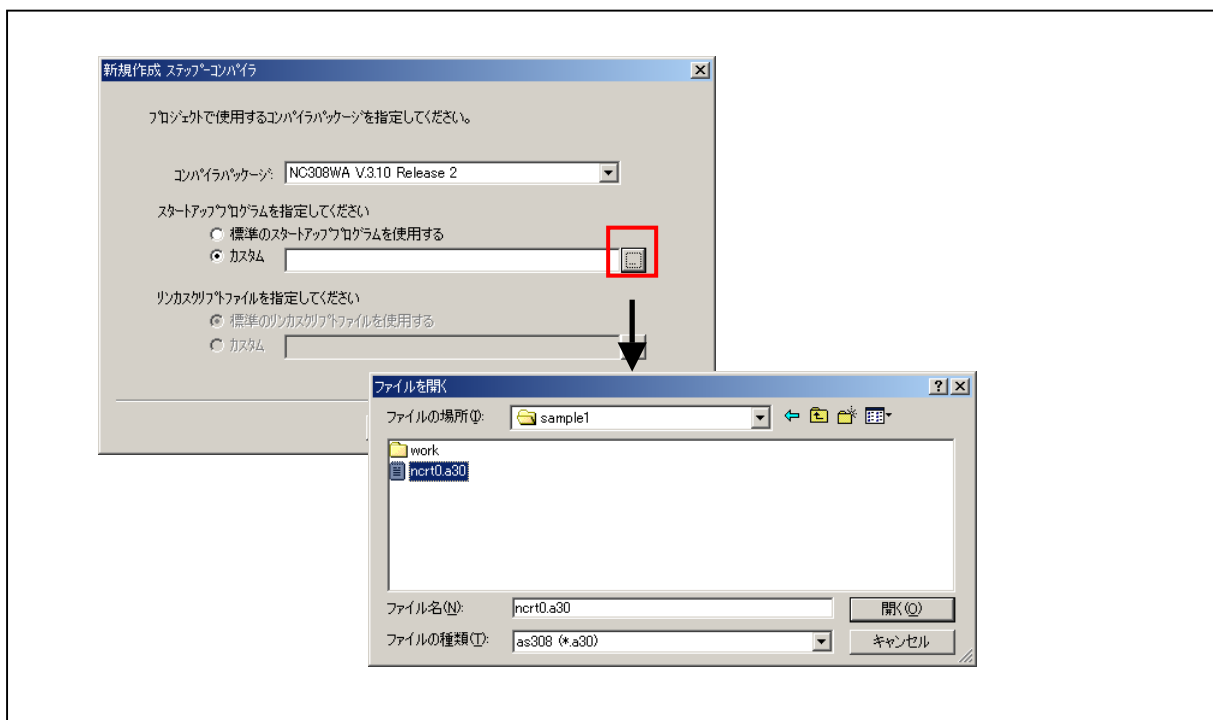


図4-5 TM：新規作成ステップーコンパイラ

ステップ完了の設定項目を確認して完了します。

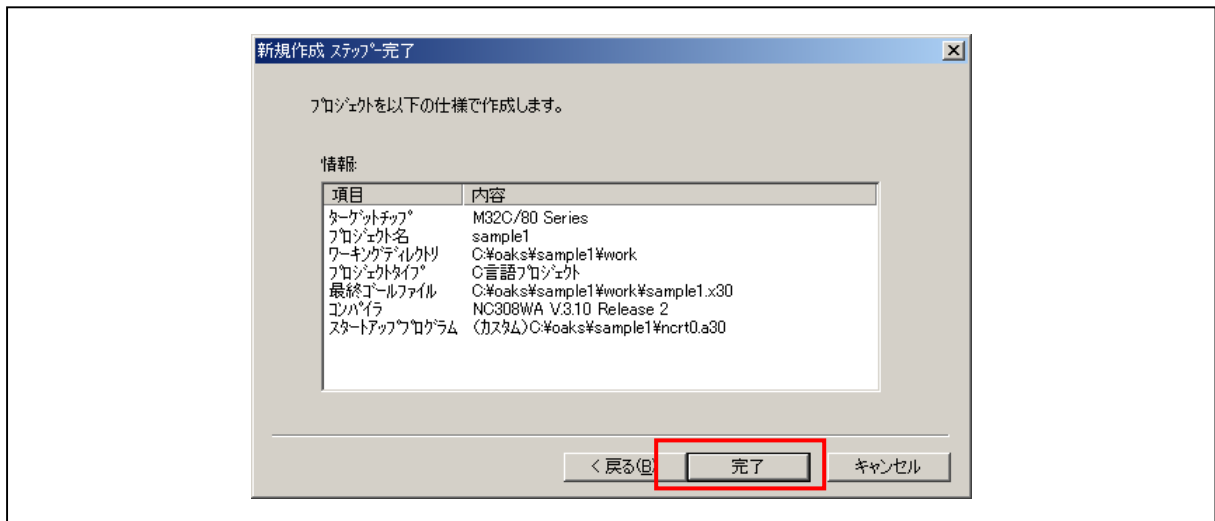


図4-6 TM : 新規作成ステップ完了

プロジェクトエディタの「+」チェックボックスをクリックして登録されているファイルを全て展開し、確認します。

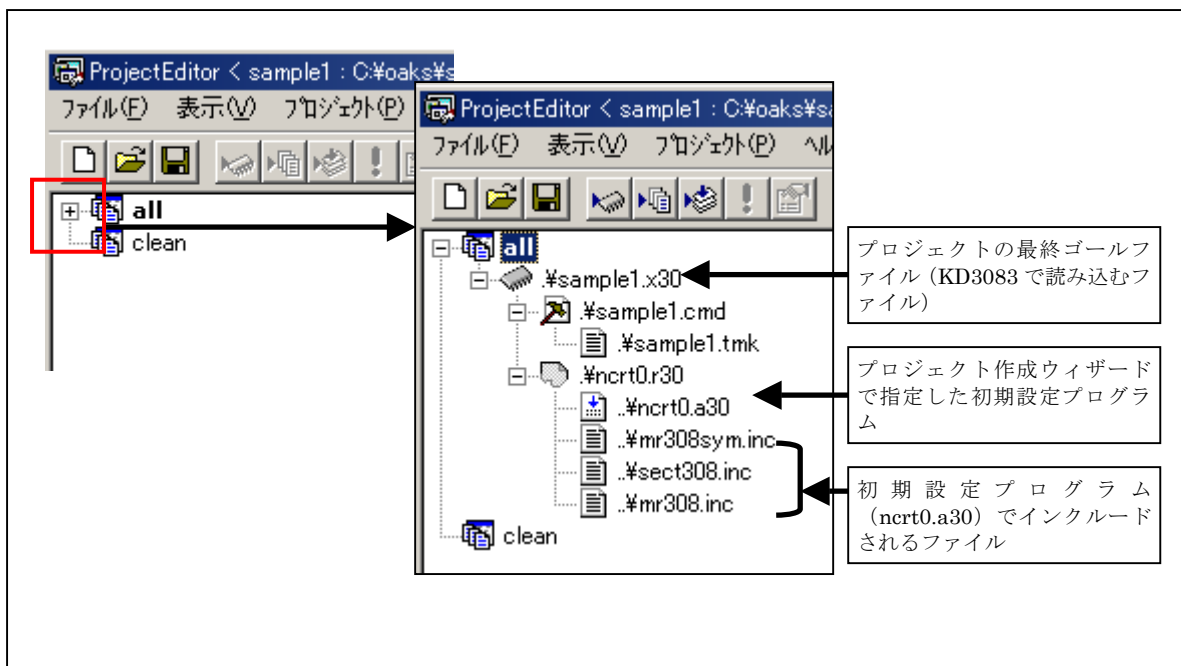


図4-7 TM : プロジェクトエディタ

### ③ファイルを追加する

プロジェクトエディタに表示されている最終ゴールファイルの`%sample1.x30`を選択して、「ファイルの追加」ボタンを押してください。追加するファイルを選んで「開く」ボタンを押してください。この手順で`mr308ent.c`、`sample1.c`を追加してください。（`ncrt0.a30`は登録済みです）

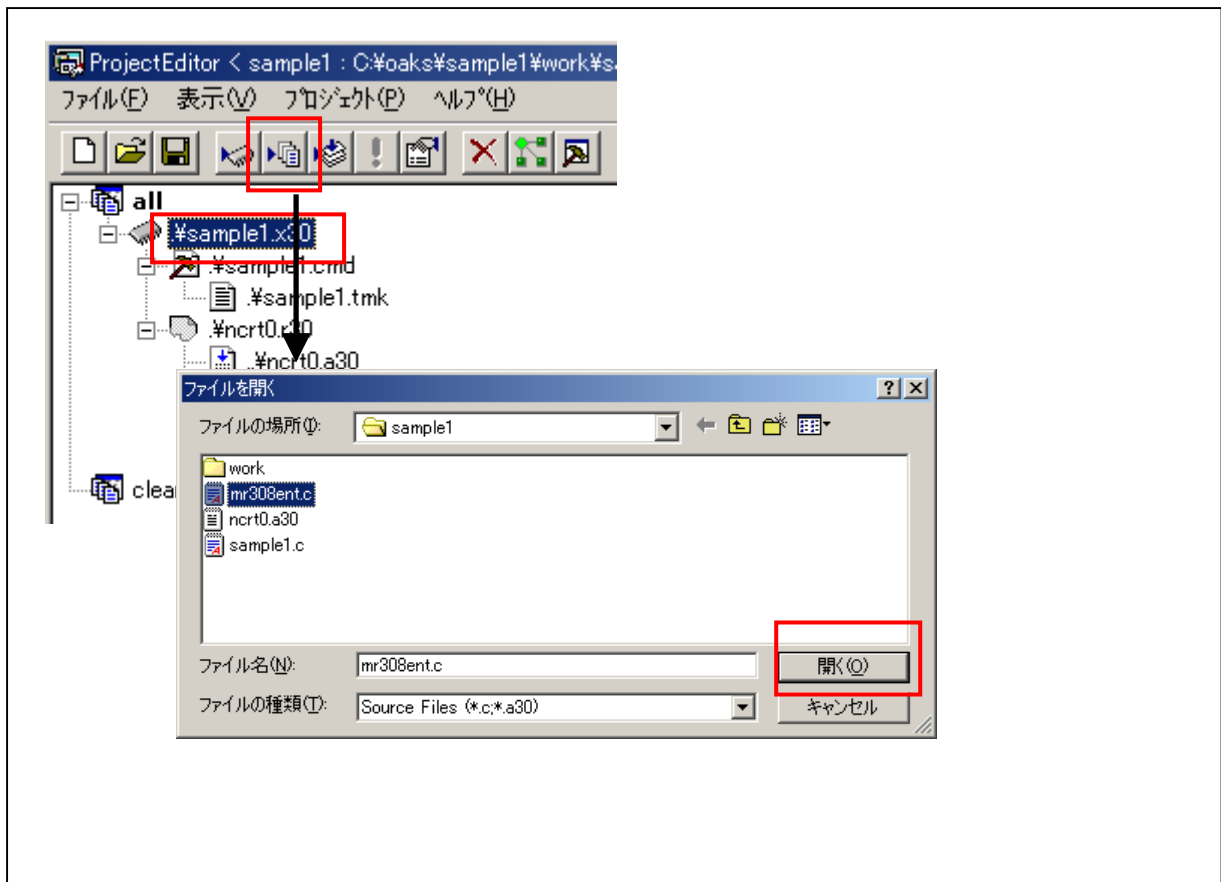


図4-8 TM：ファイルの登録

ファイルを登録したら、「依存関係の更新」ボタンを押してください。  
ソースファイルにインクルードされているファイルが表示されます。

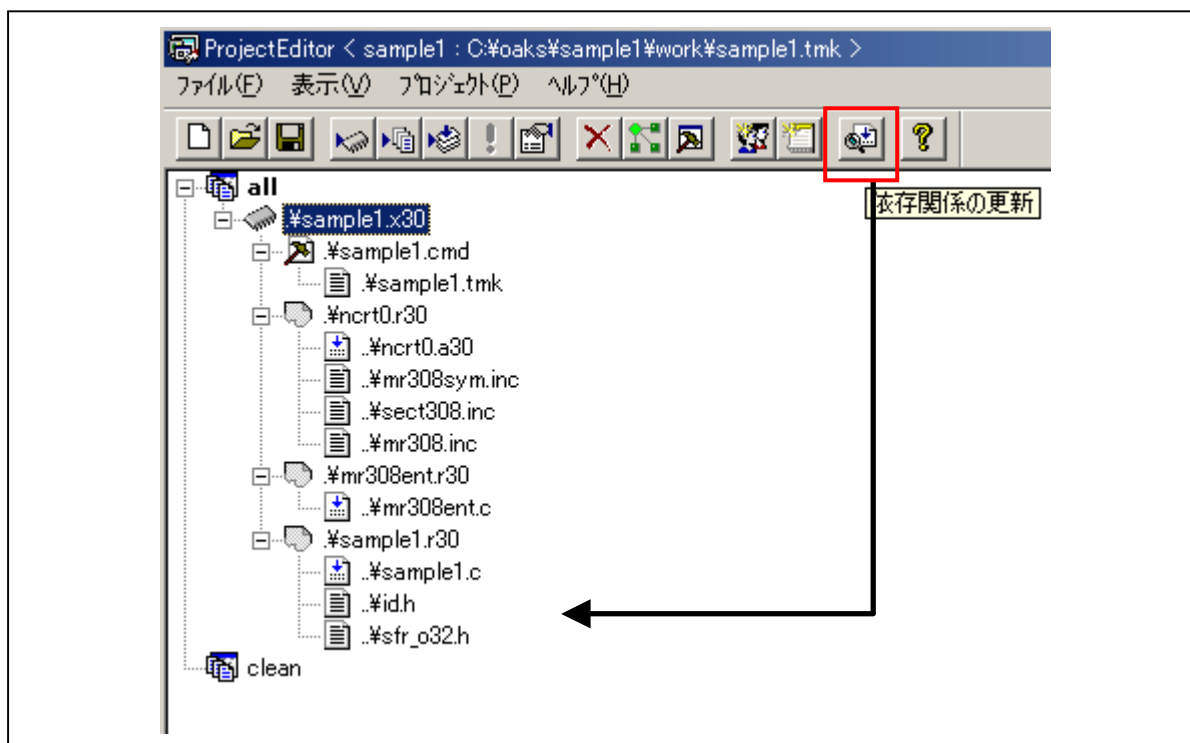


図4-9 TM : 依存関係の更新

#### ④オプションを指定する

以下のオプションを設定します。

<コンパイラオプション：-D far\_msg=1>

プロジェクトエディタの「オプションブラウザ」ボタンを押し、オプションブラウザを開きます。オプションブラウザの「CFLAGS」を選び、「編集」ボタンを押します。オプションの「-D」にチェックを入れるとパラメータ入力ウィンドウが開きます。“far\_msg=1”と入力して「OK」を押してしてください。

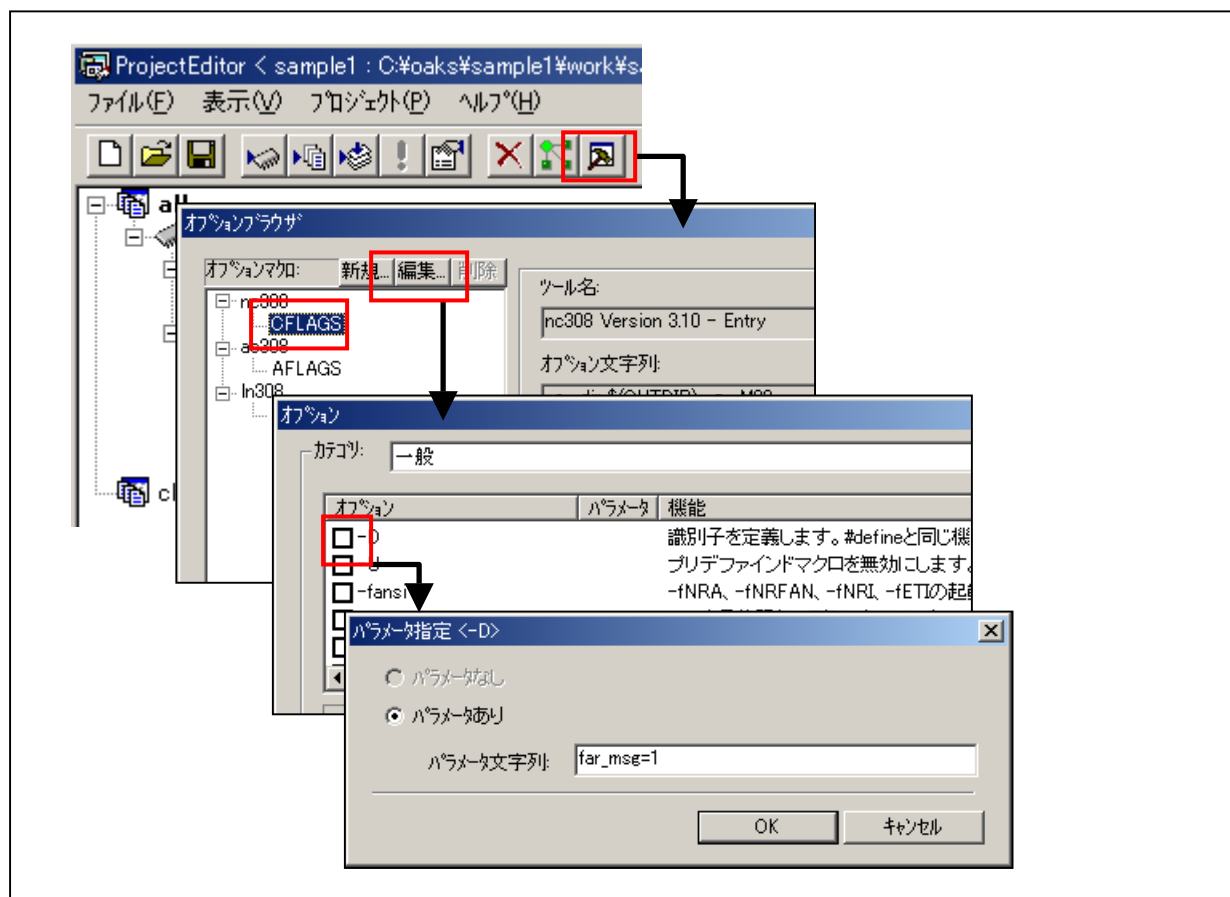


図4-10 TM：コンパイラオプションの設定

#### <リンクオプション: -L ..¥c308mrlm.lib>

オプションブラウザの「LFLAGS」を選び、「編集」ボタンを押します。カテゴリを「ライブラリの指定」として  
ください。オプションの「-L」はデフォルトで”nc382.lib”が既に設定されています。チェックを入れる  
部分を2回クリックするとパラメータ入力ウィンドウが開きます。“nc382.lib”を消して  
“..¥c308mrlm.lib”と入力して「OK」を押してください。※1

プログラム内で標準入出力関数を使用している場合は、“nc382.lib”のあとに“..¥c308mrlm.lib”と  
続けてください。※2

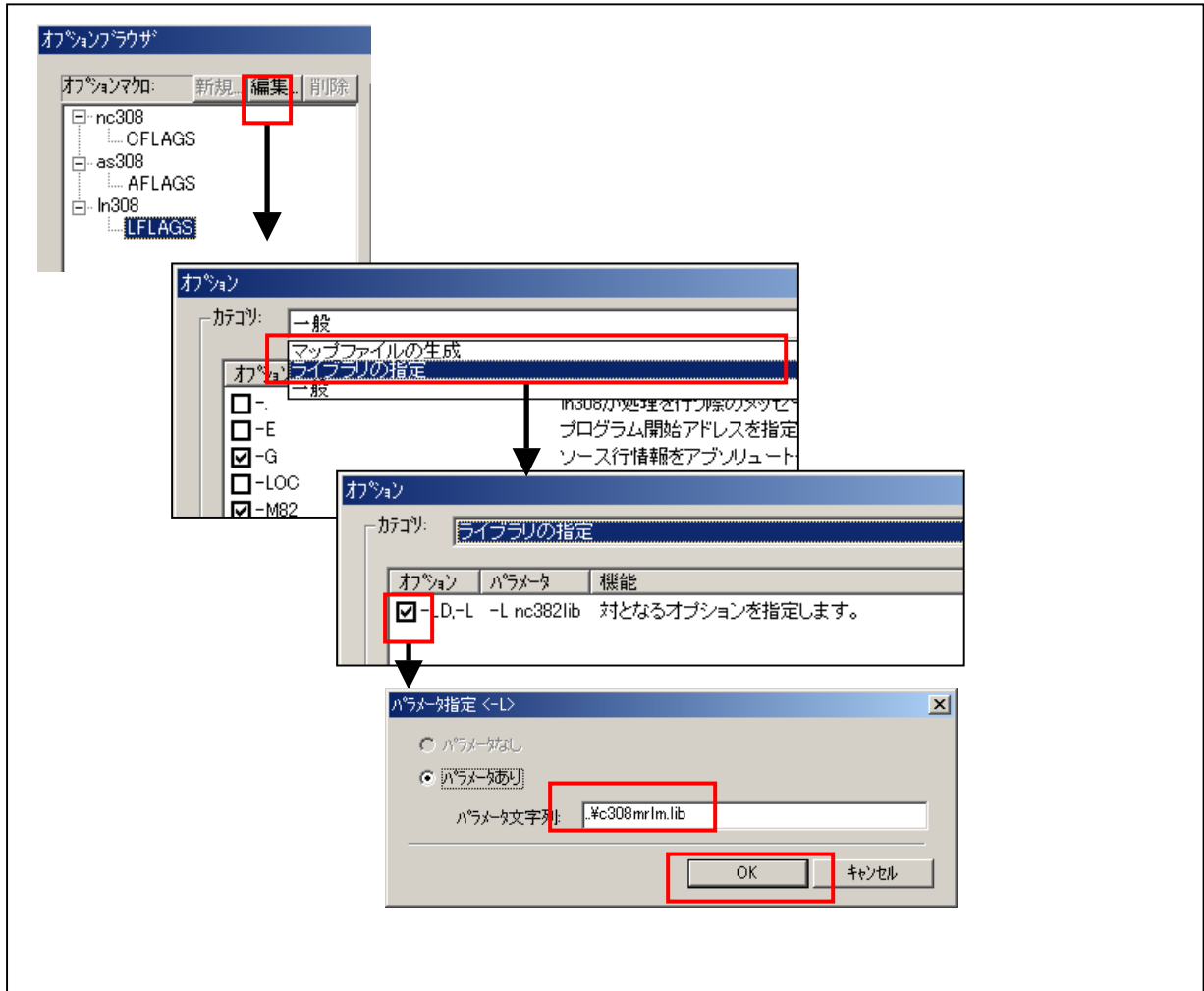


図4-11 TM: リンクオプションの設定

#### ※1

TMよりコンパイラを起動する場合、ワーキングディレクトリ（プロジェクトファイル（プロジェクト名.tmk）のあるディレクトリ）がカレントディレクトリとなります。本サンプル例ではライブラリファイルを別（一つ上）ディレクトリに置いているのでライブラリファイル指定にはパスから指定する必要があります。パスは絶対パス、相対パスどちらでも設定できます。

C:¥oaks¥sample1                   → ソースファイル、ライブラリファイルなど  
    |  
    └─ ¥work                   → ワーキングディレクトリ（カレントディレクトリとなる）

#### ※2

1つのオプションに2つ以上のパラメータを設定する場合は“,”で区切ります。

例: nc382.lib, ..¥c308mrlm.lib

### ⑤プロジェクトを保存する

プロジェクトの設定が終わったらプロジェクトエディタの「プロジェクトの上書き保存」ボタンを押します。

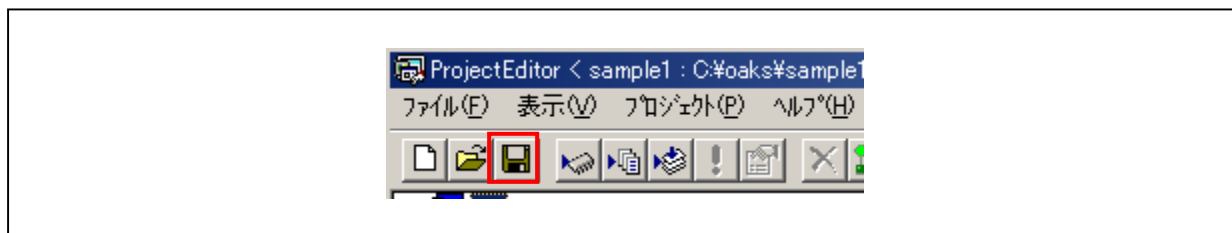


図4-12 TM : プロジェクトの上書き保存

### ⑤プロジェクトをビルドする

「ビルド」ボタンを押してビルドします。

「ビルド」は更新されたソースファイルについてのみアセンブル、コンパイルします。

「リビルド」は全てのソースファイルについてアセンブル、コンパイルします。

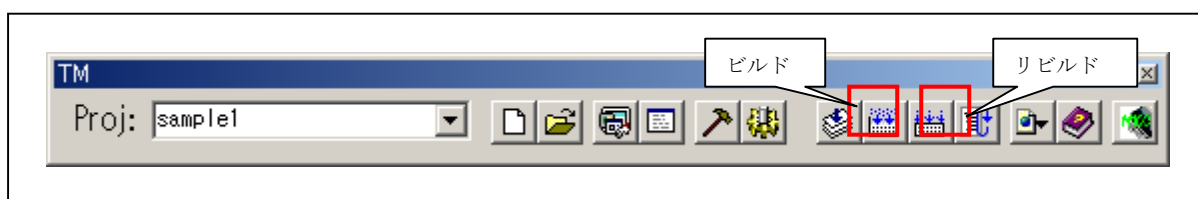


図4-13 TM : ビルドとリビルド

## 4.5 MR308 のサンプルプログラム

MR308を使用したサンプルプログラムについて説明します。

各プログラムはTMでプロジェクト設定、ビルド済みですのでKD3083を起動してオブジェクトファイル（xx.x30）をロードし、実行してください。CD-ROMからPCにコピーしたあと、すべてのディレクトリ、ファイルのプロパティの“読み取り専用”チェックをはずしてください。

### 4.5.1 周期起動ハンドラのサンプルプログラム

100msごとに周期起動ハンドラがLED点灯タスクを起動させます。LED点灯タスクでは起動されるとLEDをシフトして点灯させます。

以下にサンプルプログラムのファイル構成を示します。

表4-10 oaks32\_mr308\_tm1ディレクトリ内ファイル構成

ファイル（ディレクトリ）名	内容	
ncrt0.a30	初期設定プログラム	
mr308_sample1.c	メインのプログラム	
mr308ent.c	MR308エントリアドレス設定ファイル	
sect308.inc	メモリ空間のセクション設定のファイル	
mr308.inc	アセンブラ言語用インクルードファイル	
mr308.h	C言語用ヘッダファイル	
mr308sym.inc	シンボル定義	
id.h	ID定義ファイル	
sfr_o32.h	SFR定義ファイル	
mr308mrlm.lib	C言語インターフェースライブラリ	
<b>Work</b> （ディレクトリ） TMのビルドによって生成されたファイルが格納されます。	ファイル名	内容
	mr308_sample1.tmk	TMの設定ファイル
	mr308_sample1.x30	オブジェクトファイル
	その他の生成ファイル	

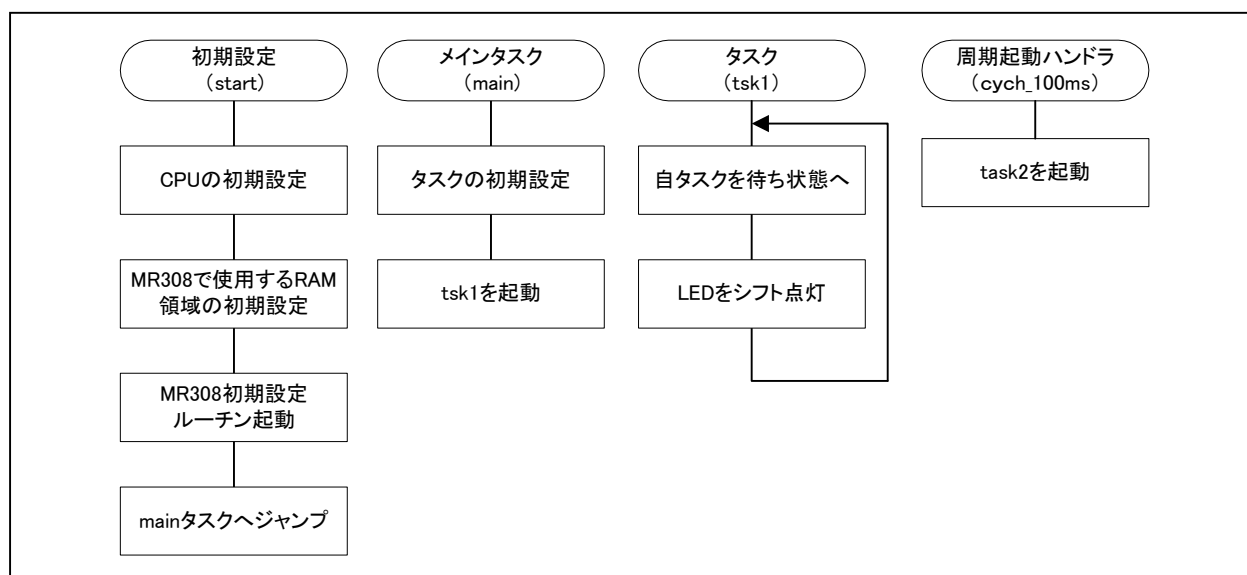


図4-14 周期起動ハンドラサンプルプログラムフローチャート

<メインプログラム リスト (mr308\_sample1.c) 1 >

```

/*-----*/
/*      MR308 を使った LED 点灯プログラム (周期起動ハンドラ) */
/*-----*/

#include <mr308.h>
#include "id.h"
#include "sfr_o32.h"

#define LedBit          0x01          /* LED シフトデータ */

#define ID_cych_100ms   ID_cyc1
#define ID_tsk1         ID_task2

int                    WupCnt;
/*****
 *   メインプログラム
 *****/

#pragma TASK main
void main(void)
{
    p1 |= 0x00;          /* P1-8bit LED 消灯*/
    pd1 |= 0xff;        /* P1-8bit OUT   */
    WupCnt |= 0x00;

    act_cyc(ID_cych_100ms, TCY_ON); /* 周期起動ハンドラをここから有効にする */
    sta_tsk(ID_tsk1, 1);          /* タスクの起動 */
    ext_tsk();
}

/*****
 *   周期起動ハンドラ (100ms)
 *****/

#pragma CYHANDLER cych_100ms
void cych_100ms (void)
{
    iwup_tsk(ID_tsk1);          /* タスクのウェイクアップ */
}

```

必ずインクルードしてください。

タスク名はデフォルトで定義されています。  
(task1,task2,...)  
プログラム内での名前を引数として使いたい場合は再定義する必要があります。

#pragma task タスク名 宣言は必ず必要です。

周期起動ハンドラはデフォルトで“OFF”となっています。必ず act\_cyc()で有効にする必要があります。

#pragma CYHANDLER ハンドラ名 宣言は必ず必要です。

<メインプログラム リスト (mr308\_sample1.c) 2 >

```

/*****
 * LED 点灯タスク
 *****/
*/
#pragma TASK tsk1
void tsk1(void)
{
    unsigned char    led_data= LedBit;    /* タスクの初期設定 */

    while(1){
        slp_tsk();    /* 周期起動ハンドラよりウェイクアップされるまで sleep 状態 */

        WupCnt++;    /* ウェイクアップされた回数をカウント (デバッグ用) */
        led_data <<= 1;    /* LED 点灯データシフト*/

        if(led_data <= 0){
            led_data = LedBit;
        }

        p1=led_data;    /*ポートへ LED 点灯データ出力 */
    }
}

```

#pragma task タスク名 宣言は必ず必要です



<MR308エントリアドレス設定プログラム リスト (mr308ent.c) >

```

/* Copyright 2003, MITSUBISHI ELECTRIC CORPORATION
 AND MITSUBISHI ELECTRIC SEMICONDUCTOR APPLICATION ENGINEERING CORPORATION
*/

// prototype declaration
// (Task & Cyclic Handler Name)

// Task
void main(void);
void tsk1(void);
void cych_100ms(void);

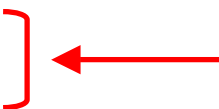
#pragma SECTION rom MR_USERROM // Don't change
void (* const _Task[16])(void) = {main, tsk1};
void (* const _Cyclic[8])(void) = {cych_100ms};

// Interval Counter
const unsigned short _D_CHCB_CNT[8] = {10};

/* Copyright 2003, MITSUBISHI ELECTRIC CORPORATION
 AND MITSUBISHI ELECTRIC SEMICONDUCTOR APPLICATION ENGINEERING CORPORATION
*/

```

タスク、ハンドラのプロトタイプ宣言  
記述した順番に ID 番号が割り付けら  
れます。ID 1 のタスク (この場合  
main) は初期起動します。



タスク、ハンドラのアドレス設定



周期起動ハンドラの周期設定  
デフォルトで1ユニットは 10ms です。  
10ms×10=100ms



## 4.5.2 イベントフラグのサンプルプログラム

周期ハンドラのプログラムと同じ動作をします。LED点灯タスクはmainで起動されるとすぐにイベントフラグ待ちになります。100msごとに周期起動ハンドラがイベントフラグをセットします。

<メインプログラム リスト (mr308\_sample2.c) 1 >

```
/*-----*/
/*      MR308 を使ったLED 点灯プログラム (イベントフラグ)      */
/*-----*/

#include <mr308.h>
#include "id.h"
#include "sfr_o32.h"

#define LedBit          0x01                /* LED シフトデータ */

#define ID_cych_100ms   ID_cyc1            /* タスク、ハンドラの定義*/
#define ID_tsk1         ID_task2

#define ID_timer_flg    ID_flg1           /* イベントフラグの定義*/
#define bit_100ms       (UINT) 0x01

UINT    flg_ptn;
/*****
 *   メインプログラム
 *****/

#pragma TASK main
void main(void)
{

    p1 |= 0x00;                /* P1-8bit LED 消灯*/
    p1 |= 0xff;                /* P1-8bit OUT      */

    act_cyc(ID_cych_100ms, TCY_ON); /* 周期起動ハンドラをここから有効にする */
    sta_tsk(ID_tsk1, 1);          /* タスクの起動 */

}


```

イベントフラグの引数となります。  
指定された型で定義するようにしてください。  
(フラグビットパターンの場合は UINT)  
指定される型については MR308 リファレンスマニュアルの該当するシステムコールの項を参照ください。  
(mr308¥doc¥mr308rj.pdf)

周期起動ハンドラは初期状態は OFF となっています。  
必ず act\_sys0 をコールしてください。

```
/*
 * 周期起動ハンドラ(100ms)
 */
#pragma CYHANDLER cych_100ms
void cych_100ms (void)
{
    iset_flg(ID_timer_flg, bit_100ms);    /* フラグのセット */
}

/*
 * LED 点灯タスク
 */
#pragma TASK tsk1
void tsk1(void)
{
    unsigned char    led_data= LedBit;    /* タスクの初期設定 */

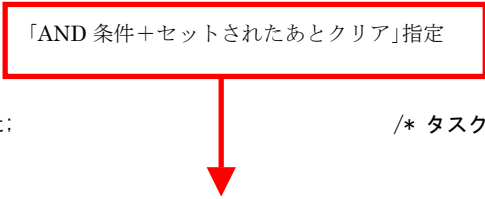
    while(1) {
        wai_flg(&flg_ptn, ID_timer_flg, bit_100ms, TWF_ANDW|TWF_CLR);
        /* フラグがセットされるまで sleep 状態 (セットされたらクリアする) */

        led_data <<= 1;    /* LED 点灯データシフト */

        if(led_data <= 0) {
            led_data = LedBit;
        }

        p1=led_data;    /*ポートへ LED 点灯データ出力 */
    }
}


```



### 4.5.3 時間管理機能（タスク遅延）によるサンプルプログラム

周期ハンドラのプログラムと同じ動作をします。8つのそれぞれ異なるLEDパターンを出力するタスクを、100msごとにとずらして起動します。

8つのタスクはLEDに出力したあと、800ms間待ち状態になります

<メインプログラム リスト (mr308\_sample3.c) >

```
/*-----*/
/*      MR308 を使った LED 点灯プログラム      */
/*-----*/
#include <mr308.h>
#include "id.h"
#include "sfr_o32.h"
#define ID_tsk1  ID_task2
#define ID_tsk2  ID_task3
#define ID_tsk3  ID_task4
#define ID_tsk4  ID_task5
#define ID_tsk5  ID_task6
#define ID_tsk6  ID_task7
#define ID_tsk7  ID_task8
#define ID_tsk8  ID_task9
#define ID_sem_p1 ID_sem1
/*****
 *   メインプログラム
 *****/
#pragma TASK main
void main(void)
{
    p1 |= 0x00;          /* P1-8bit LED 消灯*/
    p1 |= 0xff;         /* P1-8bit OUT   */

    sig_sem(ID_sem_p1); /*
    sta_tsk(ID_tsk1, 1); /*
    dly_tsk(10);        /*
    sta_tsk(ID_tsk2, 1); /*
    dly_tsk(10);
    sta_tsk(ID_tsk3, 1);

    dly_tsk(10);
    sta_tsk(ID_tsk4, 1);

    dly_tsk(10);
    sta_tsk(ID_tsk5, 1);

    dly_tsk(10);
    sta_tsk(ID_tsk6, 1);

    dly_tsk(10);
    sta_tsk(ID_tsk7, 1);

    dly_tsk(10);
    sta_tsk(ID_tsk8, 1);
```

セマフォカウンタはデフォルトは0なので、設定したい数値の回数だけ sig\_sem をコールします。

Unit time はデフォルトで10msです。  
10ms×10=100ms

```

    ext_tsk();
}

/*****
 *   タスク
 *****/

#pragma TASK tsk1
void tsk1(void)
{
    while(1)
    {
        wai_sem(ID_sem_p1);          /* セマフォ (ポート 1) を取得する */
        p1 = 0x01;                   /* ポートへ LED 点灯データ出力 */
        sig_sem(ID_sem_p1);         /* セマフォ (ポート 1) を返却する */
        dly_tsk(80);                 /* unit time 10ms×80=800ms デイレイ */
    }
}

#pragma TASK tsk2
void tsk2(void)
{
    while(1)
    {
        wai_sem(ID_sem_p1);
        p1 = 0x02;
        sig_sem(ID_sem_p1);
        dly_tsk(80);
    }
}

#pragma TASK tsk3
void tsk3(void)
{
    while(1)
    {
        wai_sem(ID_sem_p1);
        p1 = 0x04;
        sig_sem(ID_sem_p1);
        dly_tsk(80);
    }
}

```

---

以下省略します。

Tsk4 では 0x08、tsk5 では 0x10、tsk6 では 0x20、tsk7 では 0x40、tsk8 では 0x80 をポート 1 に出力します。

## 5 STCP308（オークス版）を使ったプログラム開発

OAKS32-LANBoardでSTCP308（オークス版：STCP308と呼びます）以下を使用したプログラムを開発する場合に必要な知識、手順、注意事項等について説明します。

### 5.1 STCP308 について

TCP/IPプロトコルスタックとは、TCP/IPという通信プロトコル（通信の際の決まりごと）に従って通信をするためのプログラムをライブラリとして提供しているものです。ライブラリを使用するためには、API（Application Program Interface）というプログラムインターフェースを経由します。APIは関数の形をとっています。ユーザプログラムの中でTCP、UDPの通信を行う際は、APIをコールすることで、ライブラリ内のプロトコルを使用します。STCP308が提供するAPIはITRON TCP/IP APIに準拠しています。以下にSTCP308の位置付けを示します。

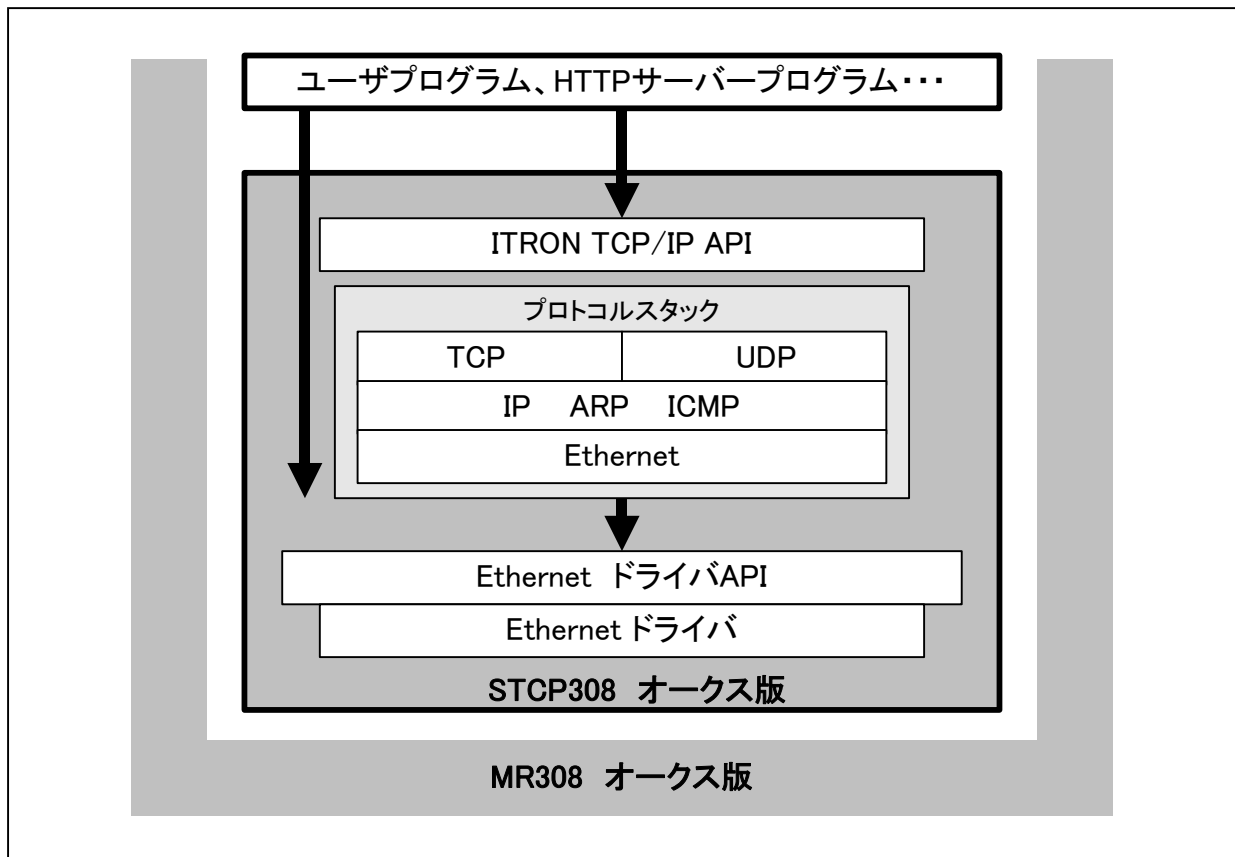


図5-1 STCP308の位置付け

## 5.2 STCP308 と OAKS32-LANBoard の H/W

STCP308はデータリンク層、物理層としてEthernetをサポートしており、OAKS32-LANBoardのH/Wにインプリメント済みのドライバを提供しています。OAKS32-LANBoardでEthernetを使用するTCP/IP通信を行うために必要なH/Wの設定などはSTCP308で提供される初期設定ルーチンやドライバで行われるので、ユーザプログラムで設定する必要はありません。

OAKS32-LANBoardではEthernetコントローラにRealtek社のRTL8019ASを搭載しています。

以下にOAKS32-LANBoardのEthernet通信のH/W構成を示します（Ethernet通信のための接続のみを表しています）。

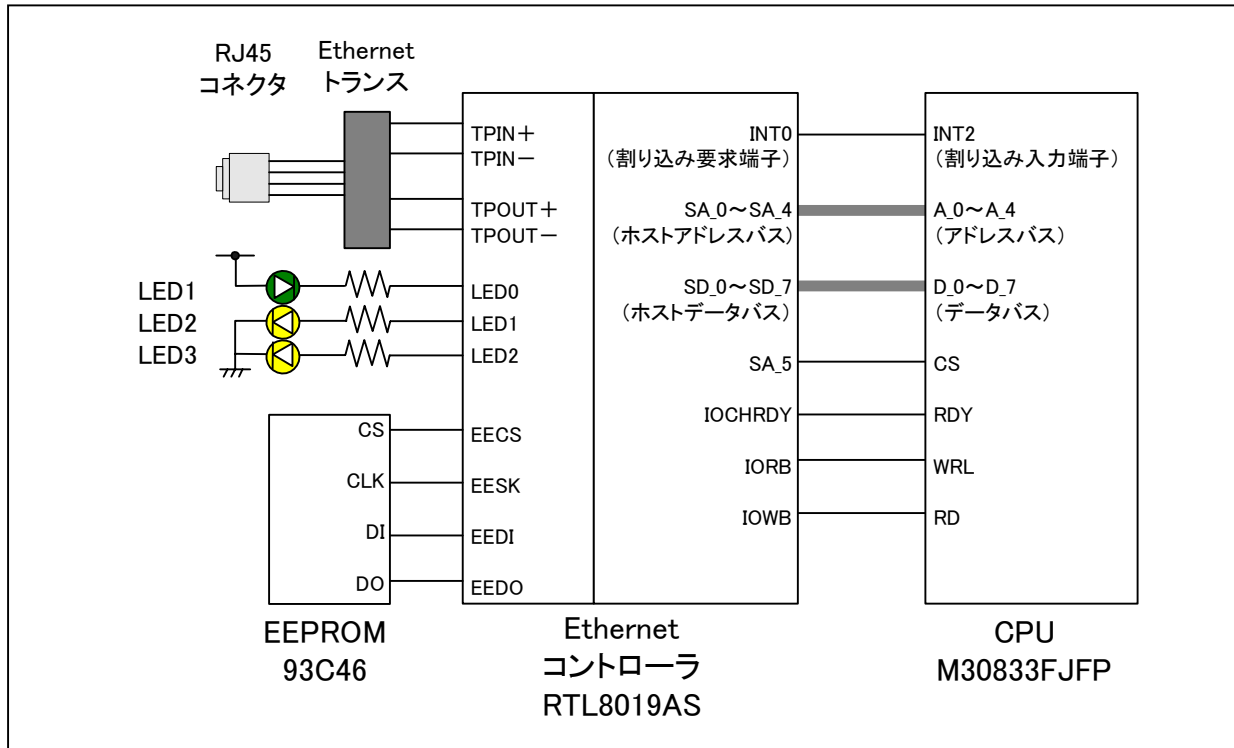


図5-2 OAKS32-LANBoard EthernetのH/W構成

### <RTL8019AS>

リアルテック社のEthernetコントローラです。接続されているEEPROMの93C46にはMACアドレスが書き込まれています。LANケーブルよりデータを受信し、あて先のMACアドレスが一致するとCPUに対して割り込み要求を出します。又、CPUより受け取ったデータをEthernet規格に合わせてLANケーブルに流します。

### <93C46>

EEPROMです。あらかじめ固定のMACアドレスが書き込まれています。ボードがショートするなどした場合、EEPROMに書き込まれたデータが消去される恐れがありますのでご注意ください。

### <LED2~4>

RTL8019ASより信号が出力されます。

LED2（緑）：リンクしている間点灯します。

LED3（黄）：データを受信している間点灯します。

LED4（黄）：データを送信している間点灯します。

## 5.2 STCP308 の機能について

STCP308（オークス版：以下STCP308と呼びます）は正規版のSTCP308（ルネサステクノロジ社製）のサポートする機能を限定し、ライブラリを機械語形式で提供したものです。

機械語ファイルにはSTCP308を使用する上で必須となるリアルタイムOS MR308も含んでいます。

正規版の場合は、ユーザプログラムで使用する機能数、オプション、アドレスなどを記述したコンフィグレーションファイルを作成し、コンフィグレータを実行してSTCP308ライブラリと整合をとるためのファイルを生成させます。ユーザプログラムのコンパイラ、アセンブラ時にこれらのファイルを組み込みます。STCP308では、コンフィグレータによる設定済みとし、STCP308ライブラリと整合をとるために必要なファイルのみを提供しています。（コンフィグレータは提供されておりません）

以下にSTCP308のコンフィグレーションの内容（抜粋）を示します。

詳細については、「STCP380オークス版 リリースノート」（CD-ROM¥STCP308¥doc¥STCP308subj.pdf）を参照してください。

### <STCP308で使用できる機能とその数>

表5-1 使用できる機能と数

UDP通信端点数		2
TCP受付口数		2
TCP通信端点数		4
TCPオプション	MSS	256
	遅延ACK送信	あり
	速度対策	あり
	シーケンス番号の初期値	1
	2MSL待ち時間	1分
	再転送タイムアウトの最大値	10分
IPアドレス		未定義
サブネットマスク		未定義
ゲートウェイアドレス		未定義
MACアドレス		EEPROMから読み出し

### 5.3 STCP308 で使用するファイル

STCP308を使ったプログラム開発をする場合、付属のCD-ROMに収められている以下のファイルを使用してください。又STCP308を使用する場合MR308も使用することになりますので、MR308に必要なファイルもあわせて使用してください。

#### <フラッシュメモリに書き込んで使用するファイル>

stcp308\_mr308mon.motはOAKS32-LANBoardのCPU内蔵フラッシュメモリに予め書き込まれています。

表5-2 mot308fileディレクトリ

ファイル名	内容
stcp308_mr308mon.mot	STCP308ライブラリ+MR308カーネル+KD3083モニタ (STC0308、KD3083と共通)
stcp308_mr308mon.mot	FlashStarter用IDファイル

#### <ユーザプログラムに組み込んで使用するファイル>

フラッシュメモリに書き込まれているSTCP308ライブラリとの整合を取るために必要なファイルです。IPアドレス、ゲートウェイアドレス、サブネットアドレス設定などはSTCP308ent.cで行います。その他のファイルは基本的に変更せずにそのまま使用します。とくにファイル内の”Don't Change”とコメントのある個所に関しては、変更するとSTCP308ライブラリが正常に動作しない場合がありますのでご注意ください。

又MR308で必要なファイルも組み込んでください。

表5-3 STCP308¥stcp308subディレクトリ

ファイル名	内容
stcp308ent.c	stcp308環境設定ファイル ユーザによる設定が必要なファイルです。
itcpip.h	ヘッダファイル ユーザプログラム（C言語）でインクルードしてください。
stcp308if.lib	STCP308C言語インターフェースライブラリ リンク時にリンクオプションを指定してリンクしてください

表5-4 MR308¥mr308subディレクトリ

ファイル名	内容
ncrt0.a30	初期設定プログラム（STCP308と共通） 必ずユーザプログラムの一番初めに実行するようにして下さい。
mr308ent.c	mr308エントリアドレス設定ファイル ユーザによる設定が必要なファイルです。
sect308.inc	メモリ空間のセクション設定のファイル（STCP308と共通） ncrt0.a30でインクルードされます。
mr308sym.inc	シンボル定義 ncrt0.a30でインクルードされます。
mr308.inc	アセンブラ言語用インクルードファイル ncrt0.a30でインクルードされます。 ユーザプログラム（アセンブラ言語）でインクルードしてください。
mr308.h	C言語用ヘッダファイル ユーザプログラム（C言語）でインクルードしてください。
Id.h	ID定義 ユーザプログラムでインクルードしてください。
c308mr1m.lib	mr308 C言語インターフェースルーチンライブラリ リンク時にリンクオプションを指定してリンクしてください。

## 5.4 コンパイル時に必要なオプション

コンパイル時に必ず以下のオプションを指定してください。

### <コンパイラオプション>

-Dオプション (パラメータ : far\_msg=1, FAR) → **-D far\_msg=1 -DFAR**

-fenable\_registerオプション → **-fenable\_register**

### <リンクオプション>

-Lオプション (指定ファイル : c308mrlm.lib, stcp308if.lib)

→ **-L ..¥c308mrlm.lib -L ..¥stcp308if.lib**

### ※-Dオプション

識別子名を定義します。

### ※-fenable\_register

レジスタ記憶クラスを有効にします。

### ※-Lオプション

リンク実行時に参照するライブラリファイル名を指定します。指定するファイル名がカレントディレクトリかあるいは環境変数 (LIB30) で指定されているディレクトリ内の場合はファイル名のみを指定します。

## 5.5 TM (Tool Manager) でプログラムをビルドするまでの手順

TM上でSTCP308を使用するプログラムを開発する場合の手順を説明します。  
ルートディレクトリCに以下のディレクトリ、ファイルを生成したものとします。

作成したディレクトリ : C:\%oaks%\sample1\work ※1  
sample2ディレクトリはソースプログラムを格納する  
workディレクトリはTMでビルド後に生成されるファイルを格納する

作成したプログラム : C:\%oaks%\sample1ディレクトリに” sample1.c” というC言語で記述したプログラムを作成したものとします。

※1プロジェクトのディレクトリ名について  
TMは空白分を含むディレクトリは指定できませんので、ご注意ください。  
例：「My Documents」、「Program Files」等は指定できません

①付属のCD-ROMのSTCP308, MR308フォルダより以下のファイルをsample2ディレクトリにコピーする。

<STCP308\stcp308subフォルダより>

- stcp308if.lib
- itcpip.h
- stcp308ent.c

<MR308\mr308subフォルダより>

- mr308.h
- c308mrlm.lib
- ncert0.a30, mr308ent.c, mr308.inc, mr308.h, sect308.inc, mr308sym.inc, id.h

表5-5 STCP308を使ったプログラムのファイル構成例

ディレクトリ	ファイル (ディレクトリ)	
C:\%oaks%\sample1	sample1.c	作成したプログラム
	ncert0.a30 mr308ent.c stcp308ent.c itcpip.h mr308.inc mr308.h sect308.inc mr308sym.inc id.h mr308.h c308mrlm.lib stcp308if.lib	コピーしたファイル
	\work	ビルドで生成されたファイルを格納するディレクトリ

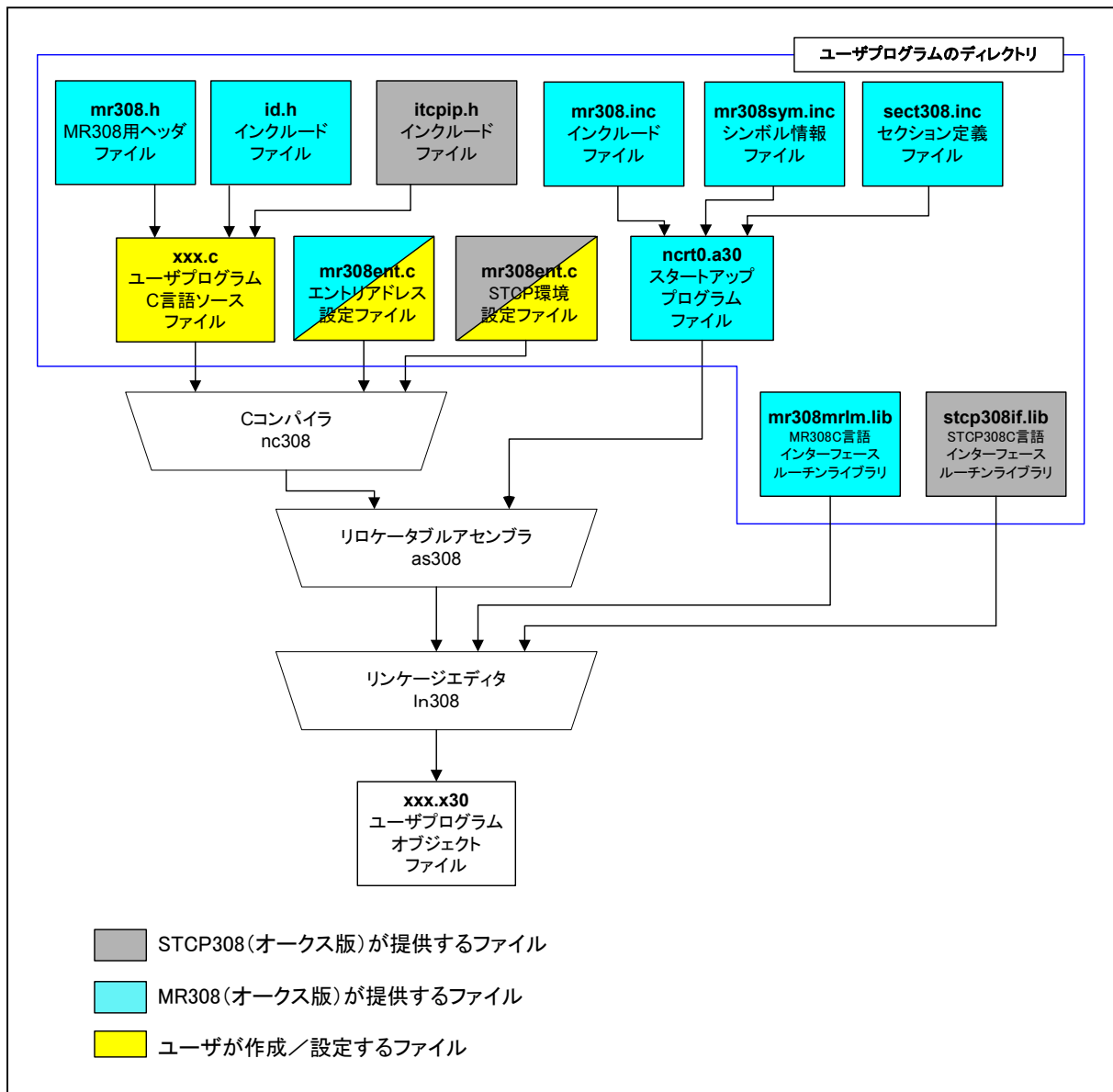


図5-3 STCP308オブジェクトファイル生成フロー図 (プログラムをC言語で記述した場合)

## ②TMを立ち上げ、プロジェクトを作成する

「New Project」ボタンをクリックし、ウィザードに従ってプロジェクトを作成します。

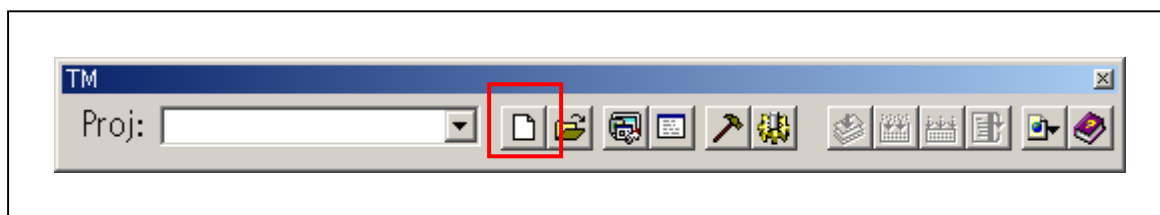


図5-4 TM New Projectボタン

ウィザードのステップ1の設定は以下のように設定してください。

- ・ターゲットチップ：M32C/80Series
- ・プロジェクト名：sample1とします。生成されるオブジェクトファイル名に反映されます。
- ・ワーキングディレクトリ：C:\%oaks%\sample1\workとします。ビルドによって生成されるファイルが格納されます。

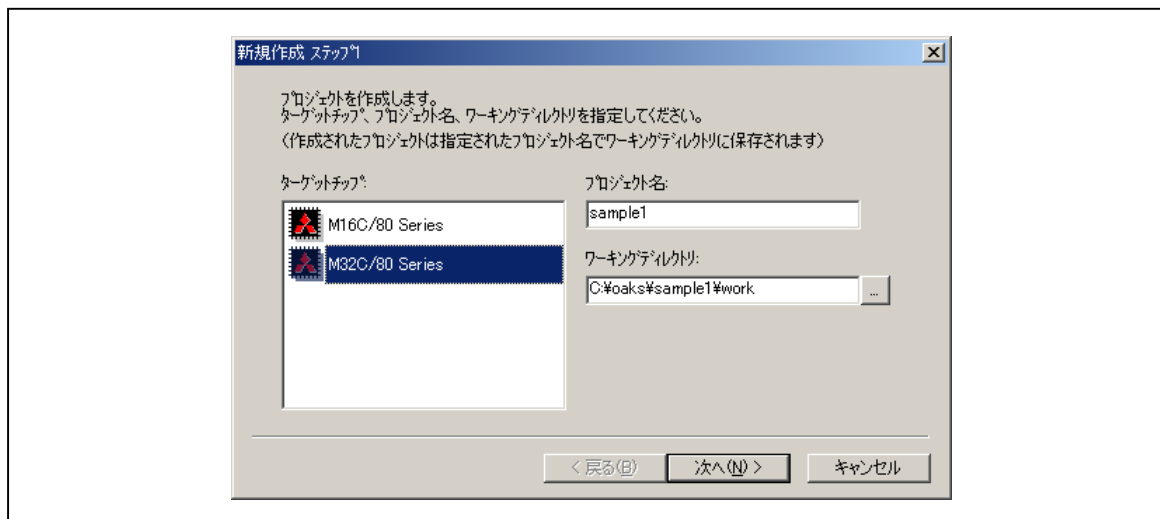


図5-5 TM：新規作成ステップ1

ステップ2の設定は「C言語プロジェクト」を選択してください。

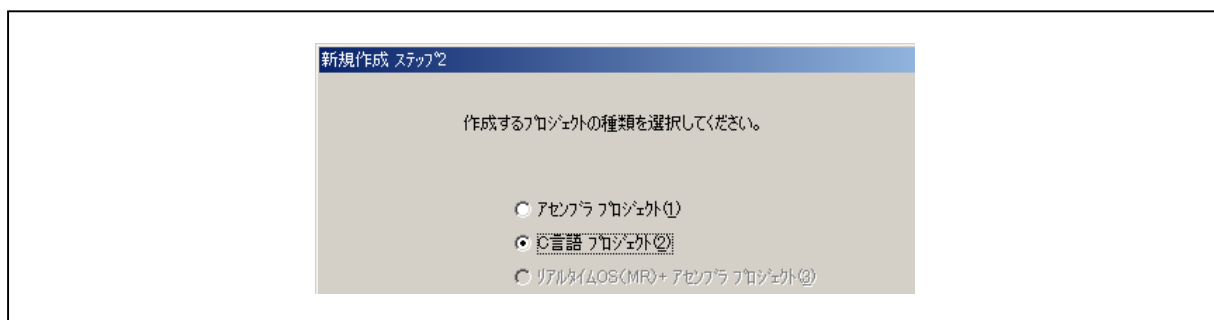


図5-6 TM : 新規作成ステップ2

ステップコンパイラでは、スタートアッププログラムを「カスタム」に指定して、初期設定プログラムを選択してください。

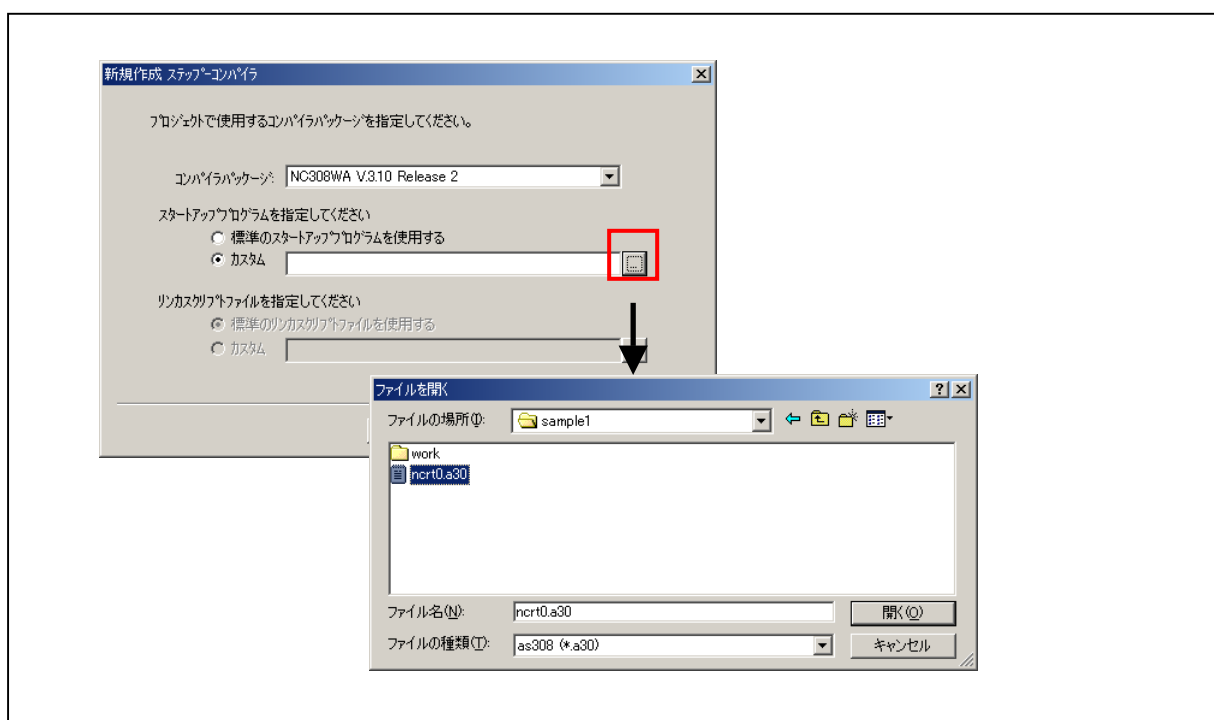


図5-7 TM : 新規作成ステップコンパイラ

ステップ完了の設定項目を確認して完了します。

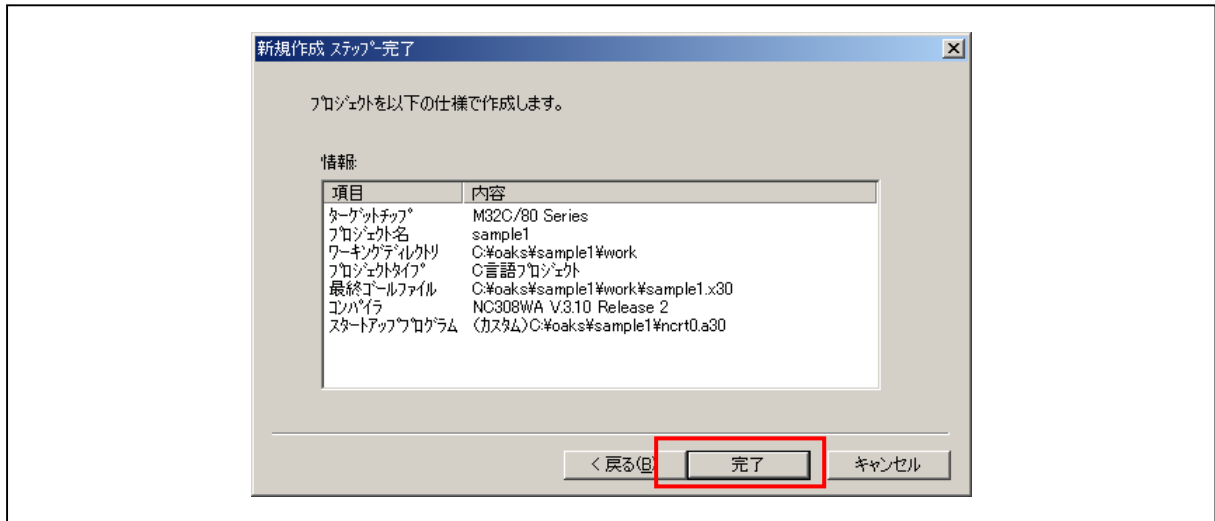


図5-8 TM : 新規作成ステップ完了

プロジェクトエディタの「+」ボックスをクリックして登録されているファイルを全て展開し、確認します。

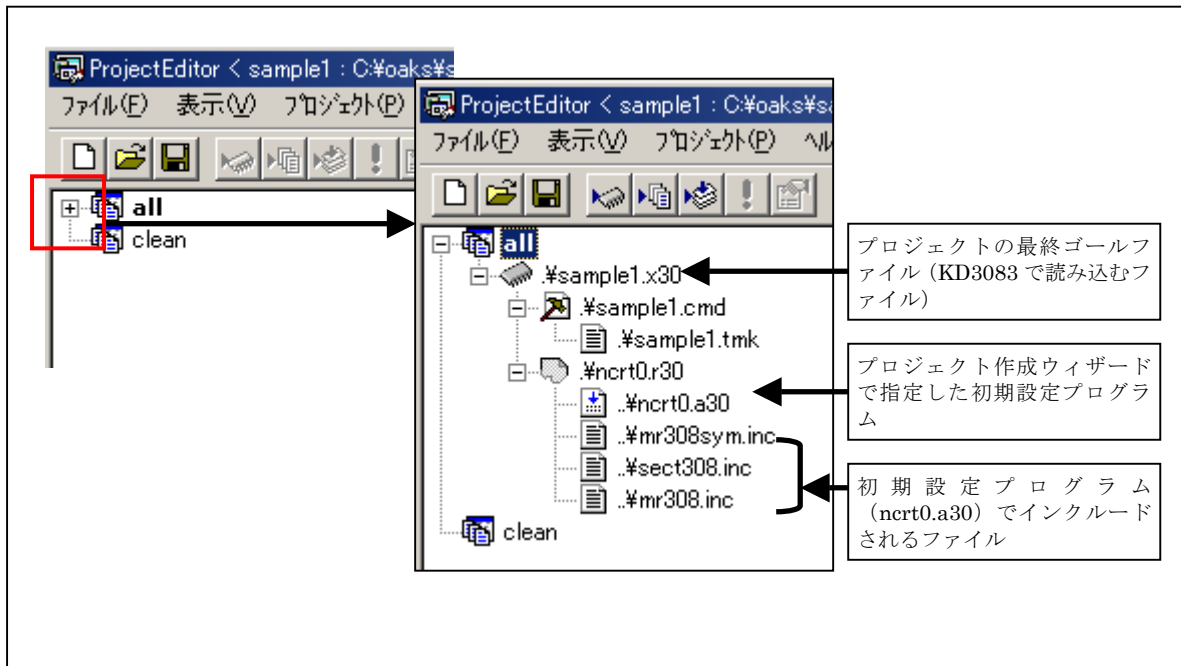


図5-9 TM : プロジェクトエディタ

### ③ファイルを追加する

プロジェクトエディタに表示されている最終ゴールファイルの`¥sample1.x30`を選択して、「ファイルの追加」ボタンを押してください。追加するファイルを選んで「開く」ボタンを押してください。

この手順で`mr308ent.c`、`stcp308ent.c`、`sample1.c`を追加してください。（`ncrt0.a30`は登録済みです）

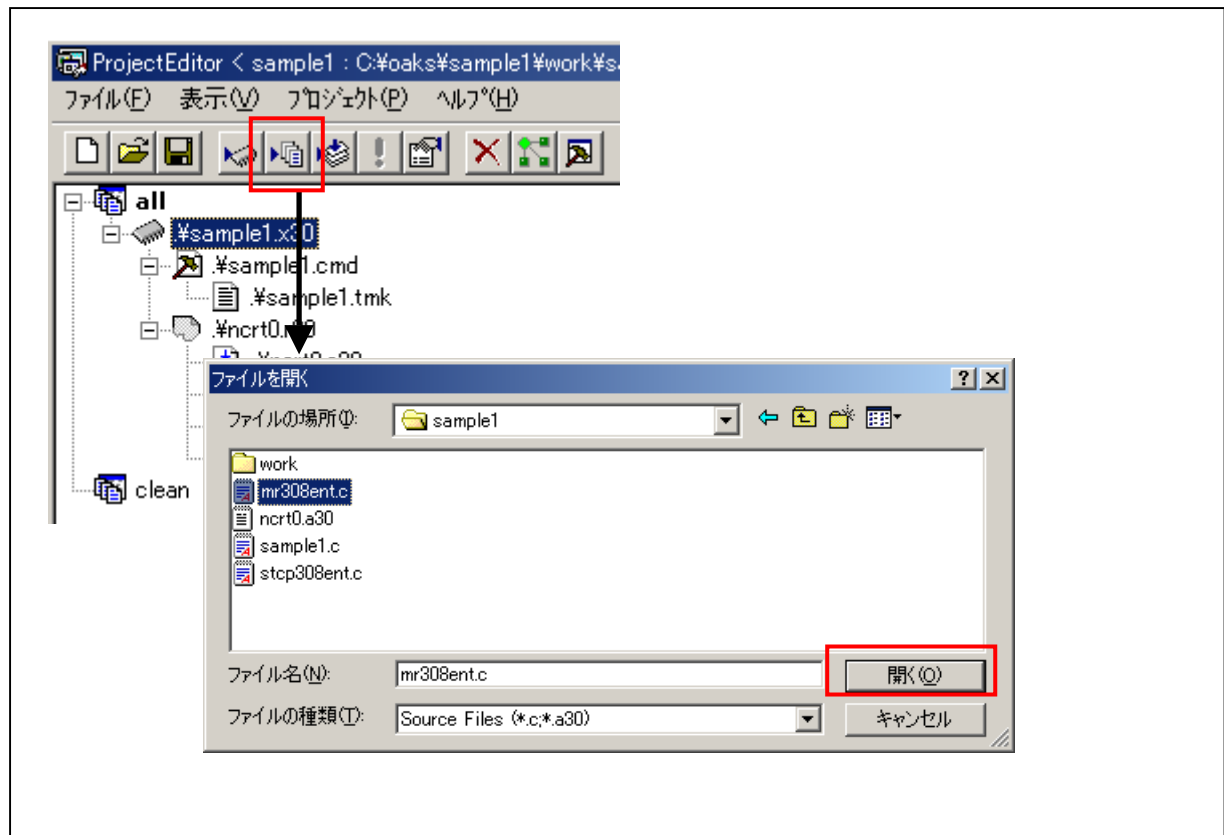


図5-10 TM : ファイルの登録

ファイルを登録したら、「依存関係の更新」ボタンを押してください。  
ソースファイルにインクルードされているファイルが表示されます。

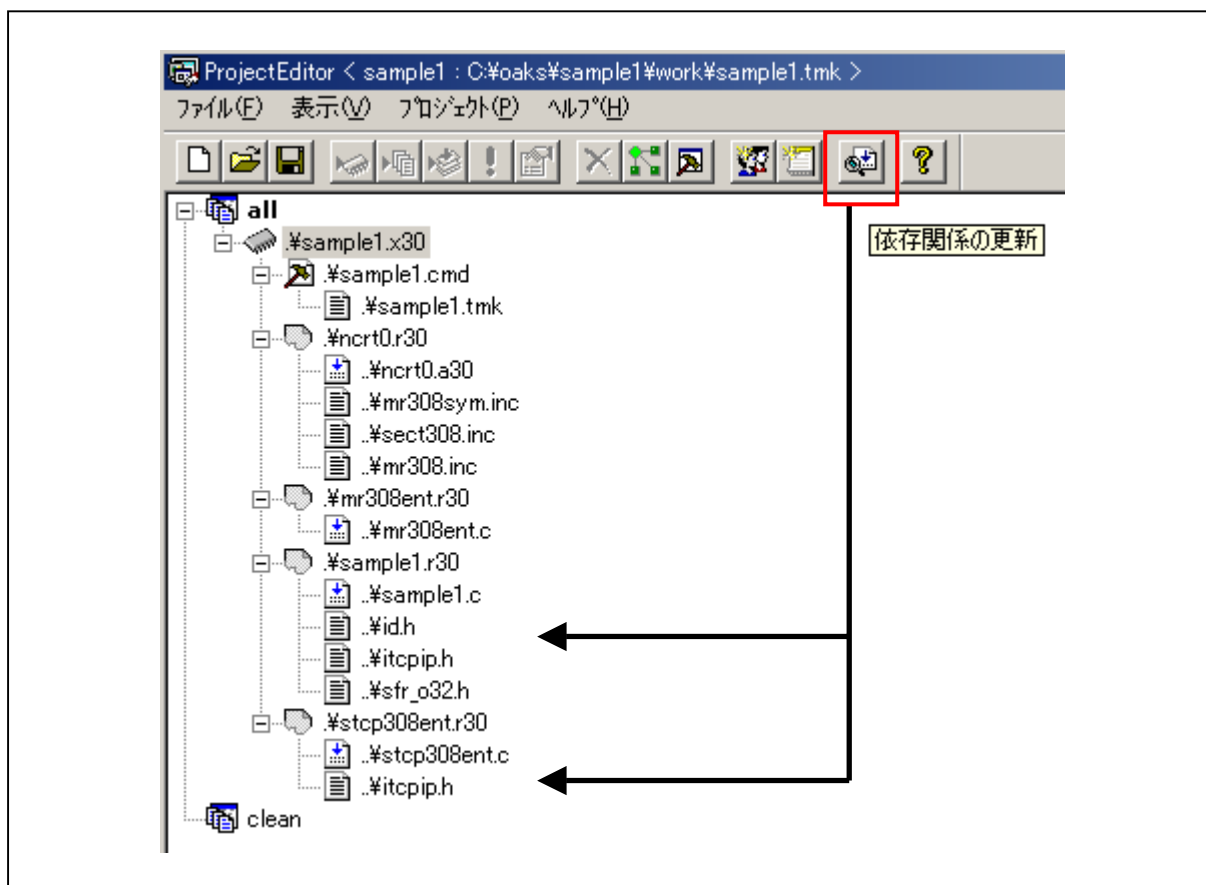


図5-11 TM : 依存関係の更新

#### ④オプションを設定する

以下のオプションを設定します。

##### <コンパイラオプション1 : -D far\_msg=1 -D FAR>

プロジェクトエディタの「オプションブラウザ」ボタンを押し、オプションブラウザを開きます。オプションブラウザの「CFLAGS」を選び、「編集」ボタンを押します。オプションの「-D」にチェックを入れるとパラメータ入力ウィンドウが開きます。“far\_msg=1,FAR”と入力して「OK」を押してください。

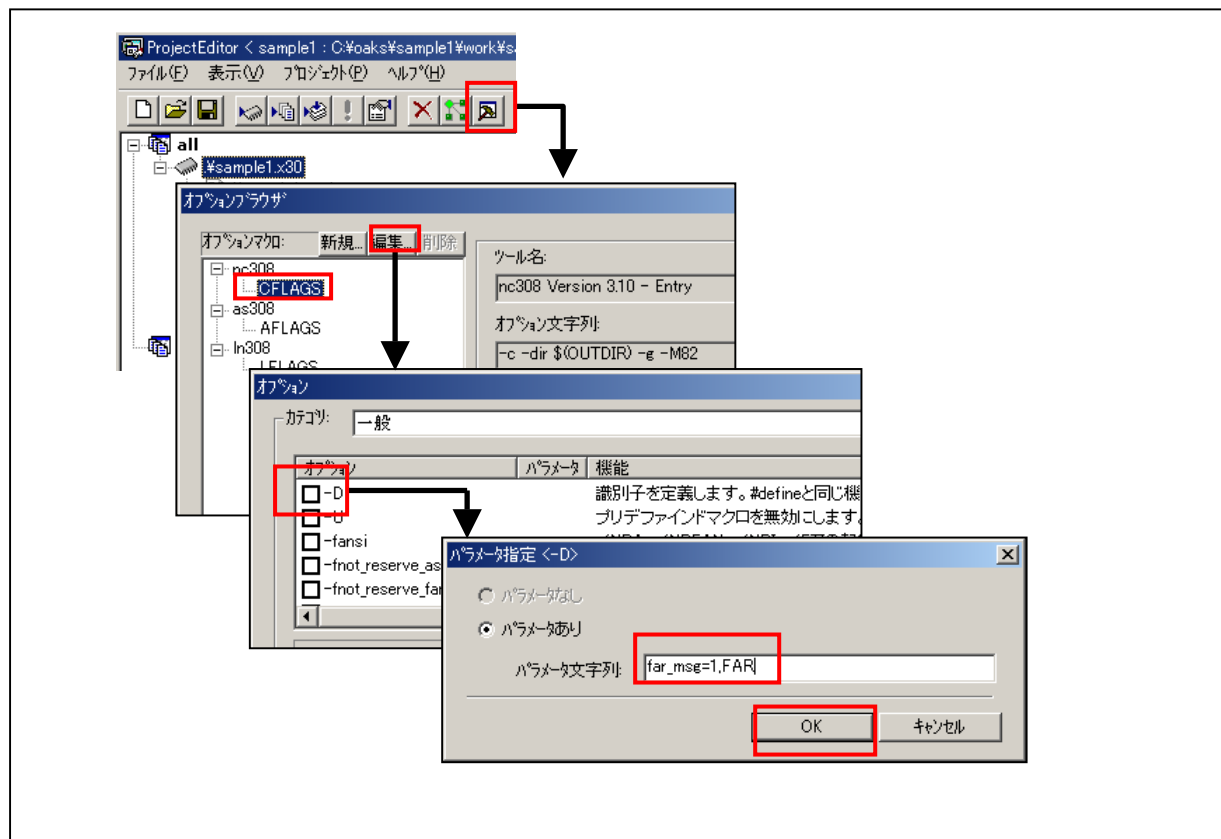


図5-12 TM : コンパイラオプションの設定 (-Dオプション)

### <コンパイラオプション 2 : -fenable\_register>

オプションブラウザの「CFLAGS」編集画面で、カテゴリを「生成コードの変更」にします。「-fenable\_register」にチェックを入れてください。

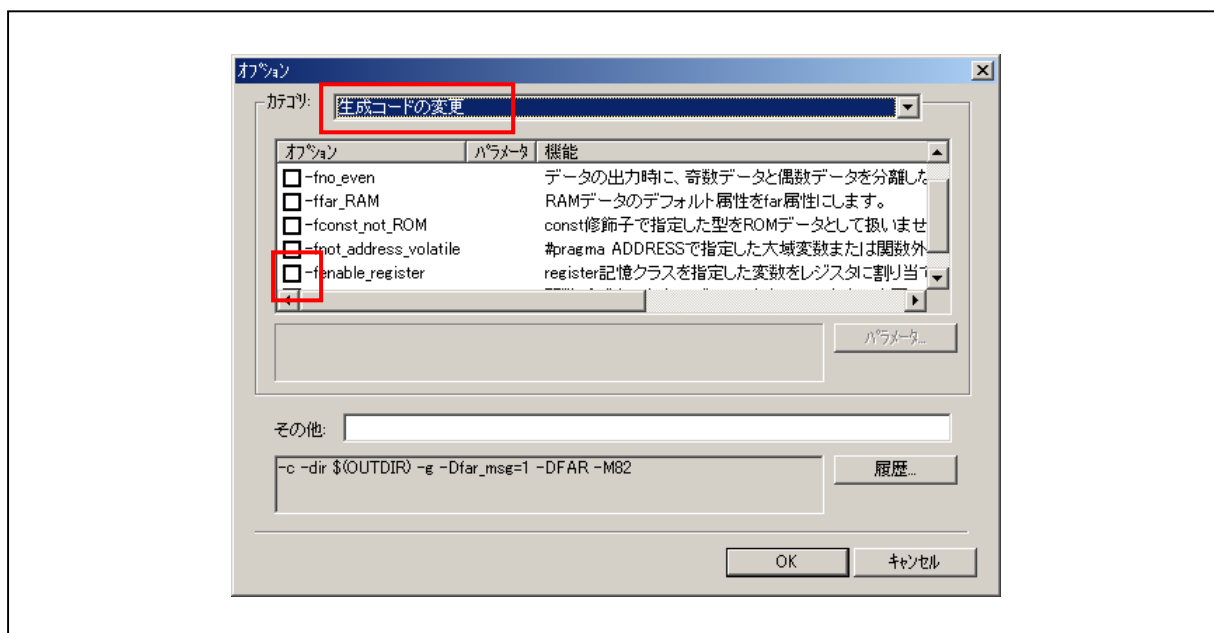


図5-13 TM : コンパイラオプションの設定 (-fenable\_register)

<リンクオプション: -L ..¥c308mrlm.lib -L ..¥step308if.lib >

オプションブラウザの「LFLAGS」を選び、「編集」ボタンを押します。カテゴリを「ライブラリの指定」として  
 ください。オプションの「-L」はデフォルトで“nc382.lib”が既に設定されています。チェックを入れる  
 部分を2回クリックするとパラメータ入力ウィンドウが開きます。“nc382.lib”を消して  
 “..¥step308if.lib, ..¥c308mrlm.lib”と入力して「OK」を押してください。※1  
 (標準関数を使用する場合は “,” で区切ってそのまま続けてください)

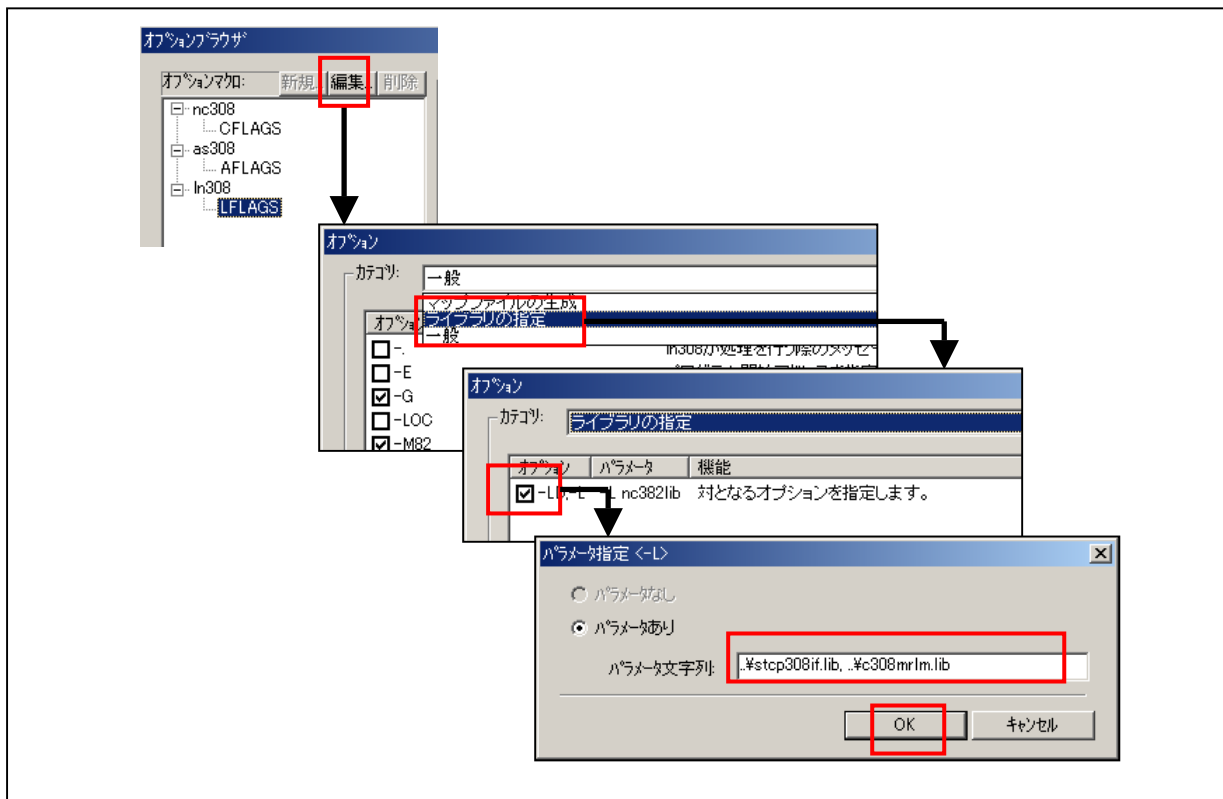


図5-14 TM: リンクオプションの設定

※1

TMよりコンパイラを起動する場合、ワーキングディレクトリ（プロジェクトファイル（プロジェクト名. tmk）のあるディレクトリ）がカレントディレクトリとなります。本サンプル例ではライブラリファイルを別（一つ上）ディレクトリに置いているのでライブラリファイル指定にはパスから指定する必要があります。パスは絶対パス、相対パスどちらでも設定できます。

C:¥oaks¥sample1                   → ソースファイル、ライブラリファイルなど  
     |  
     └─ ¥work                       → ワーキングディレクトリ（カレントディレクトリとなる）

### ⑤プロジェクトを保存する

プロジェクトの設定が終わったらプロジェクトエディタの「プロジェクトの上書き保存」ボタンを押します。



図5-15 TM : プロジェクトの上書き保存

### ⑤プロジェクトをビルドする

「ビルド」ボタンを押してビルドします。

「ビルド」は更新されたソースファイルについてのみアセンブル、コンパイルします。

「リビルド」は全てのソースファイルについてアセンブル、コンパイルします。

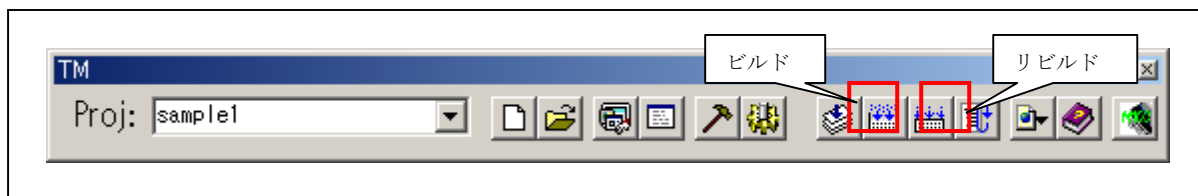


図5-16 TM : ビルドとリビルド

## 5.6 STCP308 のサンプルプログラム

STCP308を使用したサンプルプログラムについて説明します。

### 5.6.1 ping を試してみよう

L A N接続したP CよりOAKS32-LANBoardへpingコマンドを送ってみましょう。

以下にサンプルプログラムのファイル構成を示します。

表5-5 oaks32\_stcp308\_tm1ディレクトリ内ファイル構成

ファイル (ディレクトリ) 名	内容	
ncrt0.a30	初期設定プログラム	
sample_tcpip1.c	メインのプログラム	
mr308ent.c	mr308エントリアドレス設定ファイル	
stcp308ent.c	stcp308 設定ファイル	
sect308.inc	メモリ空間のセクション設定のファイル	
mr308.inc	mr308アセンブラ言語用インクルードファイル	
itcpip.h	stcp308 C言語用ヘッダファイル	
mr308.h	mr308 C言語用ヘッダファイル	
mr308sym.inc	シンボル定義	
id.h	ID定義ファイル	
sfr_o32.h	SFR定義ファイル	
c308mr1m.lib	MR308 C言語インターフェースルーチンライブラリ	
stcp308if.lib	stcp308 C言語インターフェースルーチンライブラリ	
<b>Work</b> (ディレクトリ) TMのビルドによって生成されたファイルが格納されます。	ファイル名	内容
	sample_tcpip1.tmk	TMの設定ファイル
	sample_tcpip1.tmi	
	sample_tcpip1.cmd	

<サンプルプログラムを動作させる手順>

①oaks32\_stcp308\_tm1ディレクトリごとホストPCにコピーする。

すべてのディレクトリ、ファイルのプロパティの“読み取り専用”チェックをはずしてください。

②oaks32\_stcp308\_tm1ディレクトリのstcp308ent.cにIPアドレス、ゲートウェイアドレス、サブネットアドレスを設定する。

<stcp308設定ファイル (stcp308ent.c) リスト 抜粋>

```
/* STCP308の環境設定 */
#define MY_IP_ADDR      192, 168, 0, 190    /* 自局のIPアドレス */
#define SUBNET_MASK    255, 255, 255, 0    /* サブネットマスク */
#define GATEWAY_ADDR   192, 168, 0, 1     /* ゲートウェイアドレス (全て0の場合は無効) */
```

この設定は一例です。  
環境に合わせて設定してください。

③TMでプロジェクトを開き、「依存関係の更新」をし、ビルドする。

プロジェクトは設定済みです。必ず「依存関係の更新」をおこなってからビルドしてください。

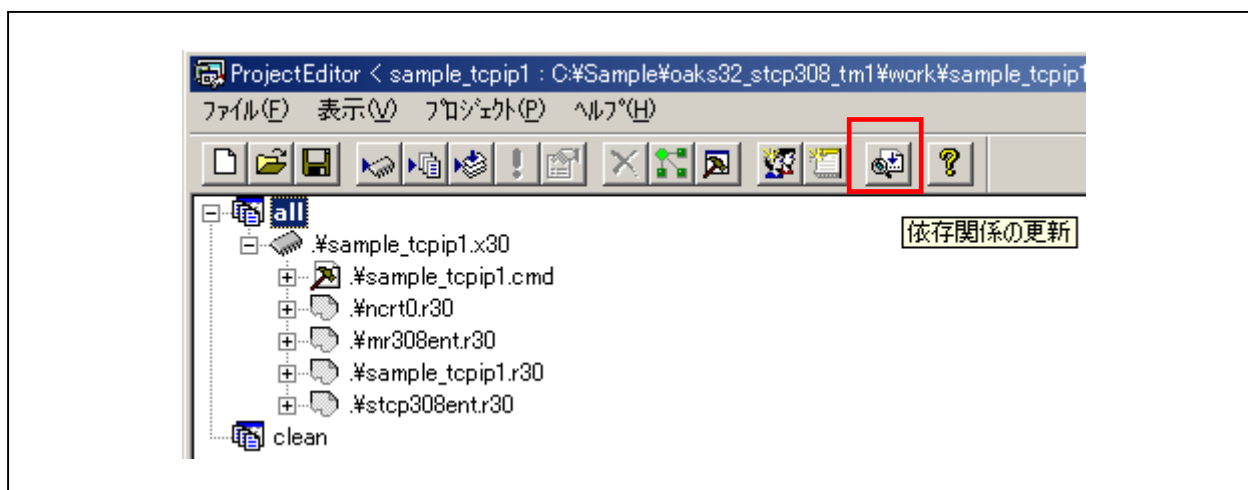


図5-17 サンプルプログラムの依存関係の更新

④OAKS32-LANBoardのセットアップをする。

RS232CケーブルでホストPCに接続します。またLANケーブルでネットワーク、あるいはPCに接続します。J1コネクタがオープンになっていることを確認して電源を入れてください。

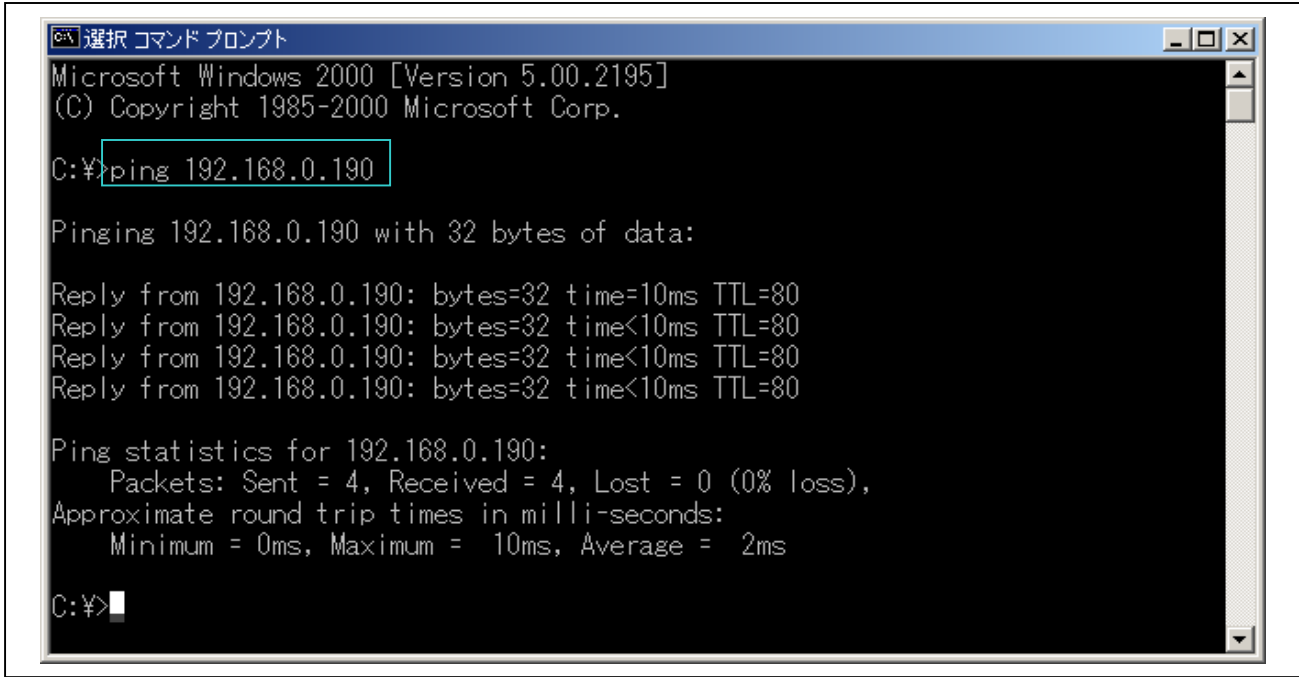
⑤サンプルプログラムを実行する。

KD3083を起動してsample\_tcpip1.x30をロードし、プログラムを実行(GO)します。あるいは、プログラムをロードしたあとKD3083を終了して、OAKS32-LANBoardをリセットし、実機動作させます。

⑥ネットワーク上のPC, あるいはLAN接続したPC上でpingコマンドを実行する。

Windowsのスタートメニューより「プログラム」→「アクセサリ」→「コマンドプロンプト」でコマンドプロンプトウインドウを開きます。

コマンド入力： >ping IPアドレス



```
選択 コマンド プロンプト
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>ping 192.168.0.190

Pinging 192.168.0.190 with 32 bytes of data:

Reply from 192.168.0.190: bytes=32 time=10ms TTL=80
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80

Ping statistics for 192.168.0.190:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms

C:\>
```

図5-18 pingコマンドの実行 (Windows2000)

### <pingコマンドとICMP>

pingは、ネットワークの接続をチェックするためのコマンドです。Windowsではping.exeが利用できます。pingはある大きさのパケットを相手方に送り、そのパケットが正常に返ってくるかを確認します（エコー要求といいます）。

pingの要求に対して応答するのはICMP（Internet Control Message Protocol）です。

サンプルプログラムの内容（次ページ）にそってみると、LANドライバの初期化が済むとLANコントローラからの割り込みが受け付けられます。この割り込み要求のベクタアドレスにはSTCP308の割り込みルーチンが割り当てられています。割り込みルーチンではこの要求の内容を調べ、各プロトコルに処理を渡します。割り込み要求のコマンドの内容がpingのICMPエコー要求の場合は、処理がICMPに渡されます。ICMPでは要求にしたがって相手先のPCにエコー応答をします。

サンプルプログラムのメインではLANドライバの初期化以降何も処理をしていません。アプリケーションの処理にかかわらず、pingのICMPエコー要求による割り込みが入れば、ICMPは応答します。

pingコマンドが成功したということは、OAKS32-Lanboradとネットワーク、あるいはPCとの接続、各H/Wの設定、各プロトコルスタック、ドライバの設定までは正しく行われているという事を意味しています。

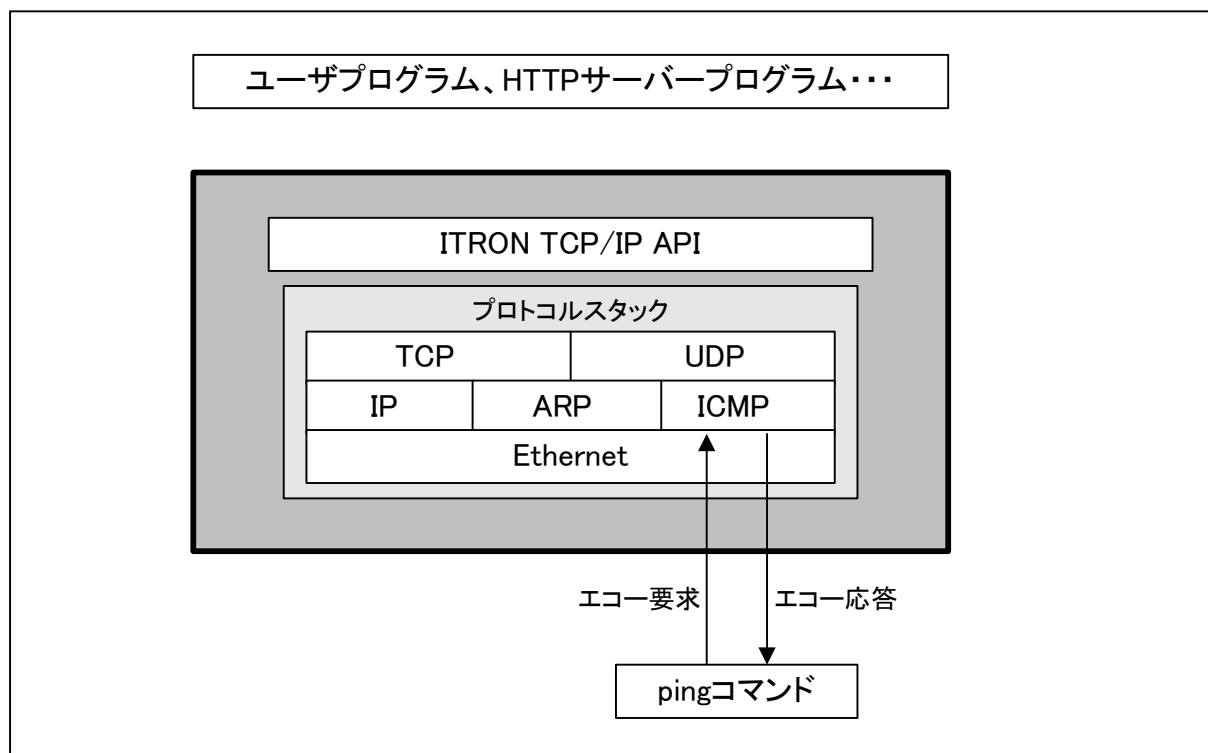
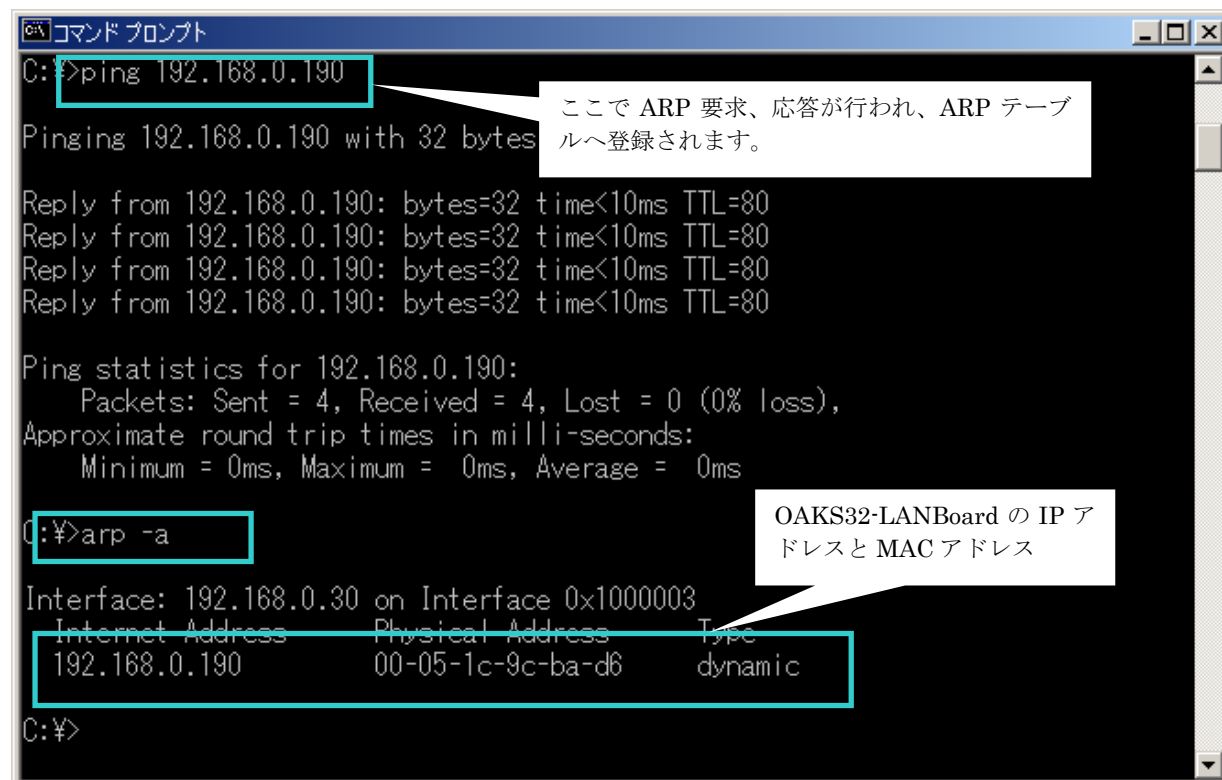


図5-19 pingコマンドとICMP

## 5.6.2 ARP コマンドで MAC アドレスを調べてみよう

ARP コマンドで OAKS32-LANBoard の MAC アドレスを調べてみましょう。  
(必ず前項の ping コマンド実行してからおこなってください)

コマンド入力: >arp -a (ARP テーブルの参照)



```
コマンドプロンプト
C:¥>ping 192.168.0.190
Pinging 192.168.0.190 with 32 bytes:
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80
Reply from 192.168.0.190: bytes=32 time<10ms TTL=80

Ping statistics for 192.168.0.190:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:¥>arp -a
Interface: 192.168.0.30 on Interface 0x10000003
Internet Address      Physical Address      Type
192.168.0.190        00-05-1c-9c-ba-d6    dynamic
C:¥>
```

ここで ARP 要求、応答が行われ、ARP テーブルへ登録されます。

OAKS32-LANBoard の IP アドレスと MAC アドレス

図5-20 ARPコマンドの実行 (Windows2000)

### <ARP (Address Resolution Protocol) コマンドとARPプロトコル>

ARP コマンドとは ARP プロトコルによって作られた ARP テーブルの内容を表示したり、修正するコマンドです。

ARP プロトコルとは、IP アドレスから MAC アドレスを求めるときに用いられるプロトコルです。OAKS32-LANBoard をネットワーク (あるいはコンピュータ) に接続した直後は、接続されたコンピュータ同士はお互いの MAC アドレスを知りません。そこではじめて通信をおこなうときは送信元より ARP 要求が出されます。

前項の ping コマンド実行時を例にあげると、ホスト PC は ping コマンド実行時に ICMP 要求をする前に、実際は ARP 要求を出しています。ARP 要求では ping コマンドで入力された IP アドレスをネットワーク上に送り (ブロードキャスト)、該当するコンピュータの MAC アドレスの返信を要求しています。OAKS32-LANBoard が該当しますので、OAKS32-LANBoard の ARP プロトコルがボード上の EEPROM に書き込まれている MAC アドレスを返信します (ARP 応答)。ホスト PC は ARP 要求の際に自分の IP アドレス、MAC アドレスも送信するので、このやりとりでホスト PC と OAKS32-LANBoard の双方がお互いの IP アドレス、MAC アドレスを知ったことになります。

一度 ARP 要求/応答のやりとりが行われると、それぞれのコンピュータは相手方の IP アドレス、MAC アドレスを自分の ARP テーブルに登録します。

つまり ping 実行後、ホスト PC の ARP テーブルには OAKS32-LANBoard の IP アドレスと MAC アドレスが登録され、OAKS32-LANBoard の ARP テーブルにはホスト PC の IP アドレスと MAC アドレスが登録されていることとなります。

<メインプログラム リスト (sample\_tcpipl.c) >

```
/*
 * サンプルプログラム : ping をためしてみよう*/
/*
 *          oaks32 support */
/*
 *          2003/5/** */
*/

#include <mr308.h>
#include "id.h"
#include "itcip.h"

#define TCPUDP_WORK_SIZ  (2048+512)
B tcpudp_work[TCPUDP_WORK_SIZ];          /* TCP/IP,UDP/IP プロトコル用のワーク領域 */

/*
 *   メインプログラム
 */
#pragma task main
void main(void)
{
    ER          ercd;
    volatile H  siz;

    /*
     *   初期化
     */

    siz = tcpudp_get_ramsize();          /* TCP/IP 用のワーク領域のサイズのチェック */
    if (siz > TCPUDP_WORK_SIZ) while(1);

    ercd = tcpudp_open(tcpudp_work);    /* TCP/IP プロトコル関連の初期化 */
    if (ercd < 0) while (1);

    ercd = lan_open(); ←
    if (ercd < 0) while (1);

    /*
     *   サンプルプログラム(メイン)
     */

    ext_tsk();
}
```

必ずインクルードしてください。

LAN ドライバの設定が終わった地点で LAN コントローラからの割り込み要求を受け付けます。

### 5.6.3 telnet のサンプルプログラム

PC上のTelnetコマンドを使って、PCよりOAKS32-LANBoardの拡張ボード上のLEDを遠隔操作するプログラムです。

サンプルプログラムは以下の構成となります。

o32_stcp308_tm2	STCP308でTELNETコマンドを実現 STCP308正規版で提供されるサンプルを本キットにインプリメントしたものです。
o32_stcp308_make2	上記サンプルをMAKEでコンパイルできるようにしたものです。 mk.batを実行してください。 MAKEファイルの設定についてはstcp308リリースノート STCP308¥doc¥stcp308subj.pdfを参照してください。
o32_stcp308_tm3	o32_stcp308_tm1のTELNETコマンドの内容を作り変えたものです。 TELNETコマンドで拡張ボードのLEDを制御します。

#### <サンプルプログラムを動作させる手順>

①～④までは前章のpingコマンドと同様です。

stcp308設定ファイル (stcp308ent.c) を以下のように設定して、ビルドしてください。

#### <stcp308設定ファイル (stcp308ent.c) リスト>

```

/* STCP308の環境設定 */
#define MY_IP_ADDR      192, 168, 0, 190    /* 自局のIPアドレス */
#define SUBNET_MASK    255, 255, 255, 0    /* サブネットマスク */
#define GATEWAY_ADDR   192, 168, 0, 1     /* ゲートウェイアドレス (全て0の場合は無効) */

.
.
.

// UDPのポート番号
const UH my_udp_port[2];

// TCPのポート番号
const UH my_tcp_port[2] = {23, 0};

```

この設定は一例です。  
ご自分の環境に合わせて設定してください。

Telnet は TCP 通信をするコマンドなので、TCP のポートのみ設定します。通常 Telnet で使用するポート番号は 23 になります。  
stcp308 ではポート番号を 2 つまで設定できますがこのプログラムでは 1 つのみ使うので、2 つめは 0 を設定します。

⑤ネットワーク上のPC,あるいはLAN接続したPC上でTelnetコマンドを実行する。

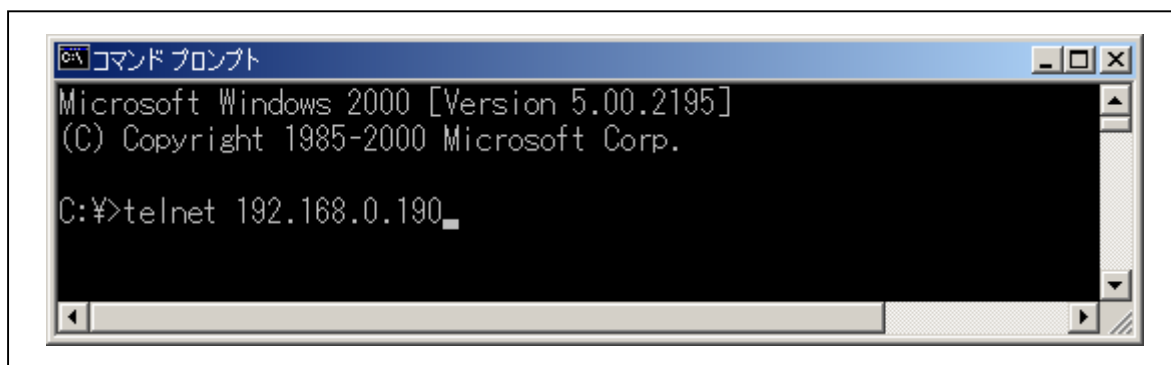


図5-21 telnetコマンドの実行

接続が確立すると、新しくウインドウが開きます。通信ポイントは4つまで開くことができます。  
(同時に4つtelnetのウインドウを開くことができます)

● o32\_stcp308\_tm2ディレクトリのサンプルプログラム

Telnetコマンドは以下のとおりです。

```
>your name?  
>your ram saize?  
>your rom size?
```

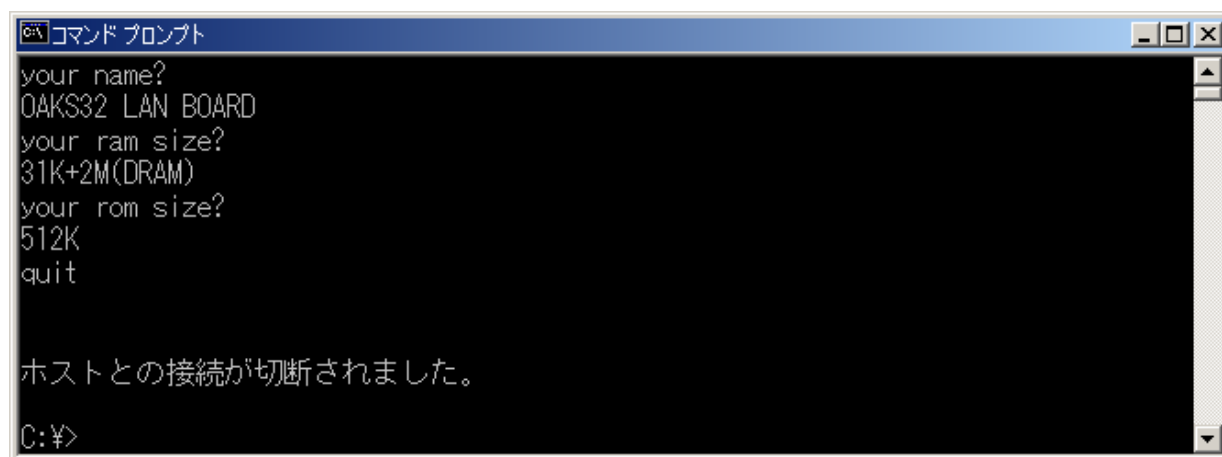


図5-22 telnet-your…コマンドの実行

● o32\_stcp308\_tm3 ディレクトリのサンプルプログラム

Telnetコマンドは以下のとおりです。

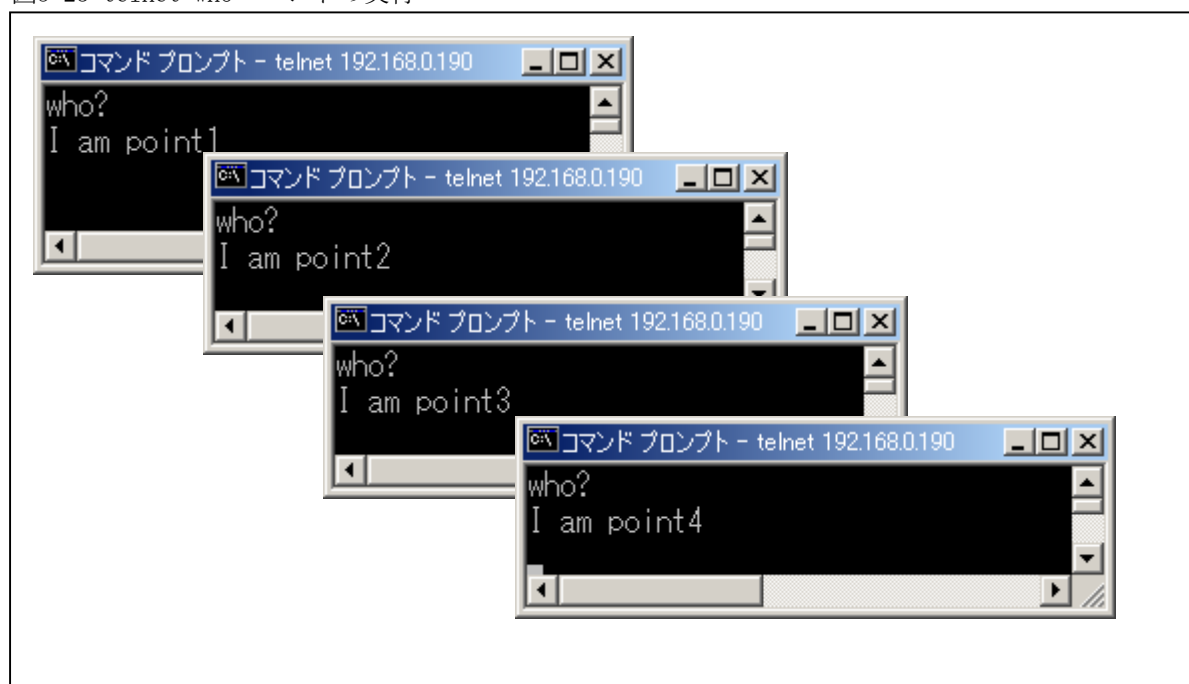
LEDコマンド：指定したLEDが点灯します。

例：>led1 →拡張ボードのLED1が点灯します。

WHOコマンド：開いているポイントのIDを表示します。

例：>who? →現在開いているポイントのIDを表示します。

図5-23 telnet-whoコマンドの実行



## 6 制限事項

**OAKS32-LANBoard**でKD3083、MR308、STCP308を使用する場合、ユーザのプログラム作成においてCPUの使用に制限事項が発生します。ご使用になる前に必ずお読みください。

### 6.1 端子の使用に関する制限事項

#### <KD3083モニタプログラムによる制限>

端子番号	機能	備考
31	TxD1 (UART1端子)	ホストコンピュータとのRS232C通信に使用します。 RS232Cドライバ/レシーバMAX232Aと接続しています。
32	RxD1 (UART1端子)	
33	CLCK1 (UART1端子)	
34	RTS1 (UART1端子)	
24	P85/NMI	NMI 割り込みは使用禁止です。 <b>OAKS32-LANBoard</b> 上でプルアップしていますが、ユーザターゲット上でLow レベルにならないように処置ください。

#### <STCP308による制限>

端子番号	機能	備考
81~88	DO (0~7)	TCP/IP通信に使用します。 EthernetコントローラRTL8019ASと接続しています。
44	RD	
46	WRL	
68~72	A (0~4)	
39	RDY	
47	CS0	
18	INT2	

#### <その他の拡張機能による制限>

※拡張機能はユーザプログラムで使用できます

端子番号	機能	備考
81~88	DO (0~7)	DRAM M5M417800と接続しています。
44	RD	
46	WRL	
51~61	A (9~19)	
40	ALE	RS232Cドライバ/レシーバMAX232Aと接続しています。
35	TxD0 (UART0端子)	
36	RxD0 (UART0端子)	
37	CLCK0 (UART0端子)	
38	RTS0 (UART0端子)	

## 6.2 周辺機能の使用に関する制限事項

### <KD3083モニタプログラムによる制限>

使用が制限される周辺機能	備考
UART1 の送信および受信割り込み	モニタプログラムとPC上のKD3083との通信に使用します。

使用するとKD8083の動作に影響を及ぼす周辺機能	備考
DMAC II のバースト転送	ユーザプログラムでDMAC II のバースト転送を使用すると通信エラーが発生する場合があります。

### <MR308による制限>

使用が制限される周辺機能	備考
タイマA0	システムタイマとして使用します。

### 6.3 メモリの使用に関する制限事項

OAKS32-LANBoardのKD3083、MR308、STCP308使用時のCPU内蔵フラッシュメモリのメモリマップを示します。ユーザ使用可能領域は（RAM 12KB、Flash ROM 445KB）となります。

MR308、STCP308未使用時でもメモリの制限は変わりません。ユーザプログラムが網掛けの領域を使用すると、プログラムが正常に書き込めない、あらかじめ書き込まれているモニタプログラム（+MR308、STCP308）が破壊される等の現象がおこる場合があります。

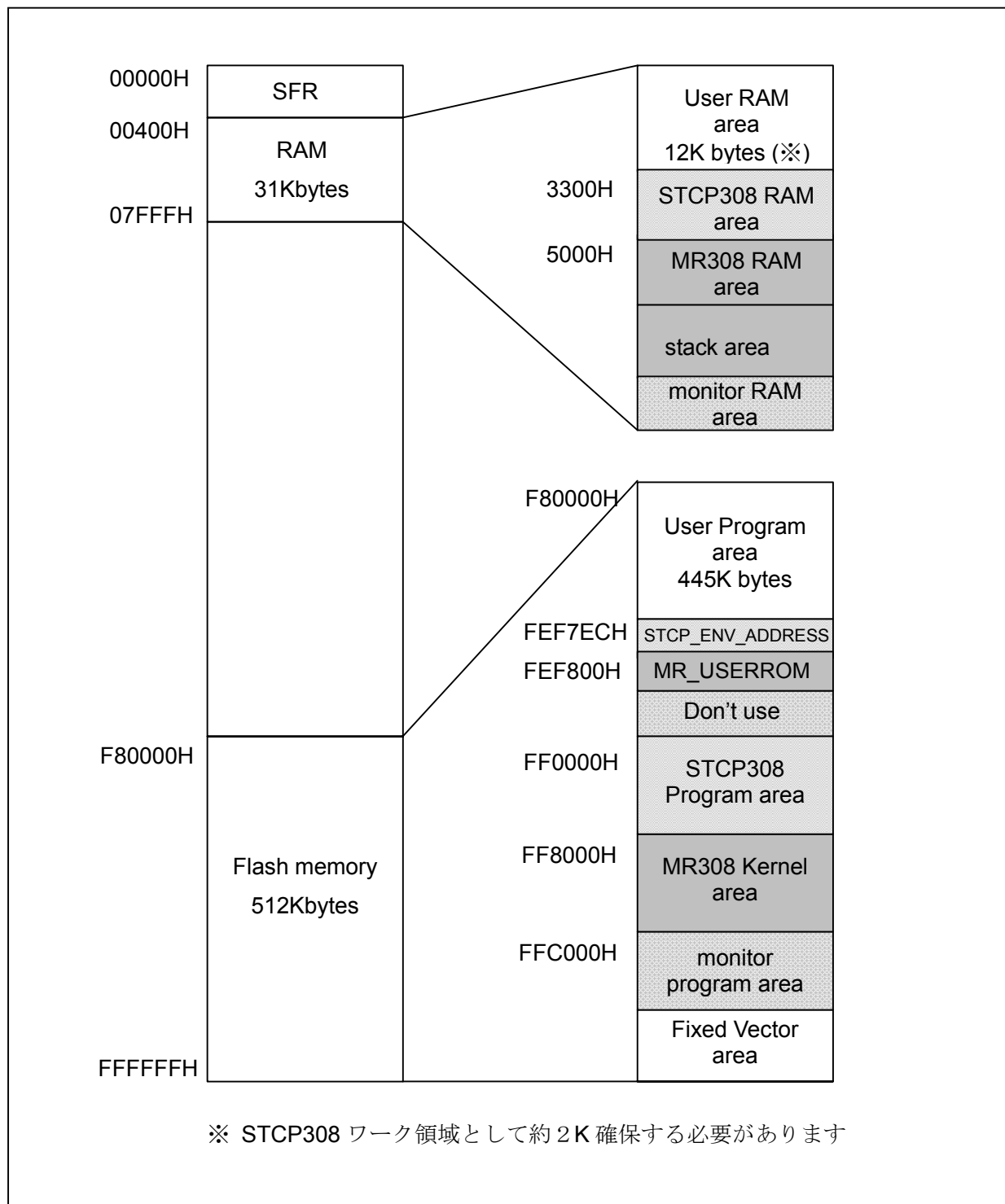


図6-1 OAKS32-LANBoardのメモリマップ（内蔵フラッシュメモリ）

OAKS32-LANBoardの外付けDRAM、およびEthernetコントローラRTL8019ASのレジスタの割り付けを示します。

DRAMの2Mバイトは全てユーザに開放しています。

DRAM、RTL8019ASを使用する場合はメモリ拡張モード、モード2に設定する必要があります。

STCP308を使用する場合は、RTL8019ASのレジスタ領域をユーザプログラムで変更しないでください。

RTL8019ASの各レジスタはSTCP308のEthernetドライバオープンAPI lan\_open()を実行することで初期化されます。

モード1		モード2	
000000h	SFR領域	SFR領域	
000400h	内部RAM領域	内部RAM領域	
008000h			
100000h			
200000h			
400000h	DRAM 2Mバイト	DRAM 2Mバイト	
600000h	DRAMイメージ	DRAMイメージ	
800000h	DRAMイメージ	DRAMイメージ	
0A00000h	DRAMイメージ	DRAMイメージ	
0C00000h	RTL8019レジスタ	RTL8019レジスタ	
	RTLイメージ	RTLイメージ	
0E00000h			
0F00000h			
0FFFFFFh	内部ROM領域	内部ROM領域	

	Page0		Page1	Page2	Page3	
	[RD]	[WR]	RD/WR	[RD]	[RD]	[WR]
C00000h	CR	CR	CR	CR	CR	CR
+1	CLDA0	PSTART	PAR0	PSTART	9346CR	9346CR
+2	CLDA1	PSTOP	PAR1	PSTOP	BPAGE	BPAGE
+3	BNDRY	BNDRY	PAR2	-	CONFIG0	-
+4	TSR	TPSR	PAR3	TPSR	CONFIG1	CONFIG1
+5	NCR	TBCR0	PAR4	-	CONFIG2	CONFIG2
+6	FIFO	TBCR1	PAR5	-	CONFIG3	CONFIG3
+7	ISR	ISR	CURR	-	-	TEST
+8	CRDA0	RSAR0	MAR0	-	CSNSAV	-
+9	CRDA1	RSAR1	MAR1	-	-	HLTCLK
+A	8019ID0	RBCR0	MAR2	-	-	-
+B	9-19ID1	RBCR1	MAR3	-	INTR	-
+C	RSR	RCR	MAR4	RCR	-	FMWP
+D	CNTR0	TCR	MAR5	TCR	CONFIG4	-
+E	CNTR1	DCR	MAR6	DCR	-	-
+F	CNTR2	IMR	MAR7	IMR	-	-
+10-17	RemoteDMA Port					
+18-1F	Reset Port					

図6-2 OAKS32-LANBoardのメモリマップ (外付けのメモリ、コントローラ)

## 6.4 レジスタ操作に関する制限事項

レジスタ操作に関する制限事項を示します。変更が禁止されているレジスタを変更した場合、ユーザプログラムの動作は保証されません。

### <KD3083モニタプログラムによる制限>

#### ①プロセッサの動作に関するレジスタ

KD3083起動時には、モニタプログラムによって以下のように初期設定されます。網掛けのレジスタは変更しないでください。その他のレジスタに関しては、必要に応じてユーザプログラムで再設定してください。

レジスタ名	設定値	注意事項
PM0 (000004h)	XXh (モニタ設定値)	モニタプログラムはKD3083起動時のステータス設定画面で設定したモードに設定します。
PM1 (000005h)	00h (リセット値)	
CM0 (000006h)	08h (リセット値)	低消費電力モードはサポートしていません。(変更不可)
CM1 (000007h)	20h (リセット値)	クロックは停止できません。(変更不可)
MCD (00000ch)	12h (モニタ設定値)	モニタプログラムは分周なしモードに設定しています。(リセット時は8分周モード)
ISP	007800h (モニタ設定値)	ユーザRAM領域内の値設定可

#### ②UART1関連レジスタ

以下のレジスタはアクセスしないで下さい。

<b>U1MR</b> (0002E8h) : UART1送受信モードレジスタ	<b>S1RIC</b> (000074h) : UART1受信/ACK割り込み制御レジスタ
<b>U1BRG</b> (0002E9h) : UART1転送速度レジスタ	<b>U1TB</b> (00036Ah) : UART1送信バッファレジスタ
<b>U1C0</b> (0002ECh) : UART1送受信制御レジスタ0	<b>U1RB</b> (00036Eh) : UART1受信バッファレジスタ
<b>U1C1</b> (0002EDh) : UART1送受信制御レジスタ1	

機能選択レジスタA0のBIT6、7 (PS0\_6、PS0\_7) のみ変更不可です。

レジスタ名	設定値	備考
PS0 (0003B0h)	10xxxxxxb	PS0_6は入出力ポート、PS0_7はUART出力 (TxD/RxD) に設定されます。

#### ③フラグレジスタ (FLG)

フラグレジスタのデバッグフラグ (D: ビット1) は変更不可です。ユーザプログラムでフラグレジスタを操作する場合はFSET、FCLR命令で行ってください。

レジスタ名	設定値	備考
フラグレジスタ	xxxxxxxxb	このフラグが” 1” のとき命令実行後にシングルステップ割り込みが発生します。

#### ④その他の注意事項

プロテクトレジスタのビット2(ポートP9 方向レジスタ、SI/03,4 制御レジスタの書込許可ビット)を“1”(許可)にした直後にモニタプログラムが動作するような場合、モニタプログラムにより任意の番地への書き込み動作が発生するため、P9 方向レジスタ書き込み許可ビットが“0”(禁止)になります。従って、以下の場合P9 方向レジスタへの書き込みはできません。

- ・書き込み許可ビットを“1”にする命令へのブレーク。
- ・書き込み許可ビットを“1”にする命令行への「Go」「Step」「Over」「Return」実行ダンプウィンドウなどからのP9 方向レジスタの操作

#### <STCP308、MR308による制限>

##### ①プロセッサの動作に関するレジスタ

MR308、STCP308で提供している初期設定プログラム(ncrt30.a30)を実行後、各レジスタは以下の設定となります。これらの設定は変更しないでください。

レジスタ名	設定値	備考
PM0(000004h)	01h	メモリ拡張モードに設定されます。 (RD/WR/BHE、マルチプレクスバスなし)
PM1(000005h)	02h	モード2に設定されます。
WCR(000005h)	dfh	外部領域2(DRAM) = 2ウエイト、外部領域3(RTL8019AS) = 3ウエイト その他の領域は3ウエイトに設定されます。
DS(00000bh)	00h	外部データバス幅は全て8ビットに設定されます。
MCD(00000ch)	12h	分周なしモードに設定されます。
ISP	5EE0h	割り込みスタックポインタは5EE0hに設定されます。

##### ②可変ベクタアドレスに関するレジスタ

MR308、STCP308で提供している初期設定プログラム(ncrt30.a30)を実行後、割り込みテーブルレジスタ(INTB: 可変ベクタアドレスの先頭番地を設定します)はFEFD00h(MR\_USERROM領域)に設定されます。ユーザRAM領域内のアドレスに変更可能ですが、ユーザRAM確保のためにも変更しないことを推奨します。

レジスタ名	設定値	備考
INTB	FEFD00h	MR_USERROM領域内に設定されます。

### ③DRAMコントローラに関するレジスタ

MR308、STCP308で提供している初期設定プログラム（ncrt30.a30）を実行後、DRAMコントローラの各レジスタは以下のように設定されます。DRAMを使用する場合は変更しないでください。

レジスタ名	設定値	備考
DRAM制御レジスタ DRAMCONT (0040h)	08h	2 ウェイト、2 MB、セルフリフレッシュ解除
DRAMリフレッシュ 間隔設定レジスタ REFCNT (0041h)	16	32mS/2048 (DRAM特性より) = 50nSx32*101 リフレッシュ間隔=BCLK周期×(リフレッシュ間隔設定ビット+1)×32

### <MR308による制限>

#### タイマA0に関するレジスタ

MR308では、システムタイマとしてタイマA0を使用します。以下の関連するレジスタ、フラグの設定は変更しないでください。

レジスタ名	設定値	備考
カウント開始フラグのビット0 TABSR, TAOS (0340h)	1	カウント開始
アップダウンフラグのビット0 UDF, TA0UD (0344h)	0	ダウンカウント
タイマA0レジスタ TAO (0346, 0347h)	36h 63h (初期値)	カウント値
タイマA0モードレジスタ TAOMR (0356h)	40h	タイマモード、ゲート機能なし、カウントソースを8分周

### <STCP308による制限>

#### INT2関連レジスタ

イーサネットコントローラからの割り込みはINT2で受け付けられます。STCP308で提供しているAPI（tcpudp\_opn() : STCP308ライブラリオープン）を実行後、各レジスタは以下の設定となります。これらの設定は変更しないでください。

レジスタ名	設定値	備考
INT2IC(00009Ch)	xx010111b	割り込み優先レベルは7、立ち下がりLエッジ、エッジセンスに設定されます。

外部割り込み要因選択レジスタのBIT3 (IFSR2) のみ変更不可です。

レジスタ名	設定値	備考
IFSR(00031Fh)	xxxxx0xxb	割り込み要因は片エッジに設定されます。

## 6.5 割り込みに関する制限事項

### <KD3083モニタプログラムによる制限>

#### ①固定ベクタテーブルに配置される割り込み

固定ベクタテーブル上の割り込みはリセット以外は使用禁止です。各固定ベクタテーブルにはモニタプログラムがダミー処理 (REIT 命令等) を格納しています。

割り込み要因	ベクタテーブル番地	モニタプログラム内の仕様
未定義	FFFDCh ~FFFDh	使用禁止 (※1)
オーバーフロー	FFFE0h ~FFFE3h	使用禁止 (※1)
BRK 命令	FFFE4h ~FFFE7h	使用禁止
アドレス一致	FFFE8h ~FFFEbh	使用禁止
シングルステップ	FFFECh ~FFFEFh	使用禁止
監視タイマ	FFFF0h ~FFFF3h	使用禁止
DBC	FFFF4h ~FFFF7h	使用禁止
NMI	FFFF8h ~FFFFbh	使用禁止
リセット	FFFFCh ~FFFFFh	使用可

(※1) : 未定義、オーバーフロー割り込みはUND、INT0命令で発生します。  
ユーザプログラム内で本命令を使用しないでください。

#### ②可変ベクタテーブルに配置される割り込み

以下の割り込みは使用しないでください。また該当する可変ベクタテーブルは変更しないでください。

使用が制限される割り込み要因	備考
UART1の受信割り込み	モニタプログラムとPC上のKD3083との通信に使用します。
DAMC IIの転送完了割り込み	

### <MR308による制限>

以下の割り込みは使用しないでください。また該当する可変ベクタテーブルは変更しないでください。

使用が制限される割り込み要因	備考
タイマA割り込み	MR308ではシステムタイマとしてタイマAを使用しています。
ソフトウェア割り込み 割り込み番号：63~55	INT命令の割り込みを、システムコールのために予約しています。 ユーザプログラムで使用する場合は66~55以外の割り込み番号を使用してください。

### <STCP308による制限>

以下の割り込みは使用しないでください。また該当する可変ベクタテーブルは変更しないでください。

使用が制限される周辺機能	備考
INT2割り込み	EthernetコントローラRTL8019ASからの割り込みを受け付けます。

## 6.6 命令に関する制限事項

### <KD3083モニタプログラムによる制限>

ユーザプログラムで、以下の命令を使用しないでください。

使用が制限される命令	備考
UND、INTO命令	命令実行により未定義、オーバーフロー割り込みが発生します。

ユーザプログラムで以下のストリング転送命令を使用した場合、KD3083の操作に注意が必要です。

使用（評価）に注意が必要な命令	備考
SCMPU, SIN, SMOVF, SOVB, SMOVU, SOUT, SSTR	命令実行中はモニタプログラムとホストPC上のKD3083が通信できません。

ストリング転送命令をKD3083で評価する場合、フリーランモードでストリング転送命令以降の命令にブレークポイントを設定し、充分時間が経過してから（ストリング転送命令が終了後）ストップしてください。

## 6.7 KD3083 に関する制限事項

### 6.7.1 ブレークポイントに関する制限事項

●LDC 命令の直後の命令にはブレークポイントを設定しないでください。LDC 命令の直後の命令にブレークポイントを設定してもブレークはかかりません。

●INT 命令上でのブレークによる停止はできません。ブレークを行った場合は、それ以降のGO コマンドは使用できません。

<例>

```

NOP
NOP
NOP
INT #3 ← ブレーク  割込み要因からのGOはできません
NOP
NOP
```

### 6.7.2 ストップモード、ウェイトモードに関する制限事項

ストップモード、ウェイトモードは使用できません。ユーザプログラム中ではストップモード、ウェイトモードに遷移しないようにしてください。

### 6.7.3 ユーザプログラムのリアルタイム性について

#### ●SamplingRun (サンプリング) モード

サンプリングモードでは、Go 実行時およびCome 実行時にユーザプログラムの実行状態を定期的に監視します。そのため、ブレークなどによるユーザプログラムの停止を検出することができます。通常のデバッグを行うときに選択してください。

#### ●FreeRun (フリーラン) モード

フリーランモードでは、Go 実行時およびCome 実行時にユーザプログラムの実行状態を監視しません。そのため、ユーザプログラムのリアルタイム性は保証されますが、ブレークなどによるユーザプログラムの停止を検出できません。したがって、ユーザプログラムが停止しても、KD3083 はGo 実行およびCome 実行動作を停止しません。KD3083 を停止させるには、STOP ボタンを押してください。ユーザプログラムのリアルタイム実行を行いたいときに選択してください。

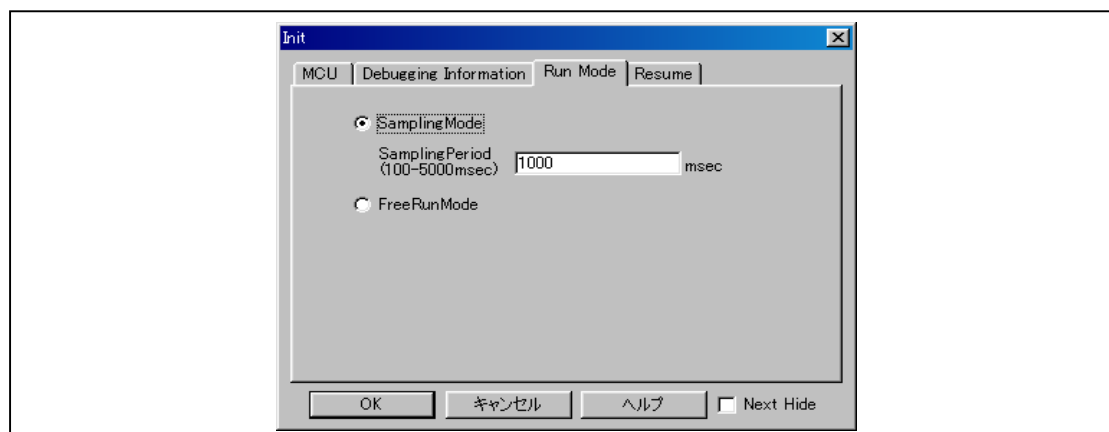


図6-5 Run Mode 設定画面

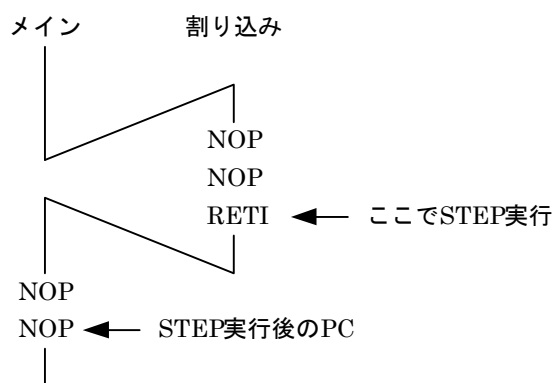
### 6.7.4 例外的なステップ実行について

以下の命令をステップ実行した場合、通常の動作と異なります。

#### ①REIT 命令、JMPS 命令、JSRS 命令

REIT 命令をSTEP 実行すると、割り込み復帰後の命令を1 命令実行して停止します。

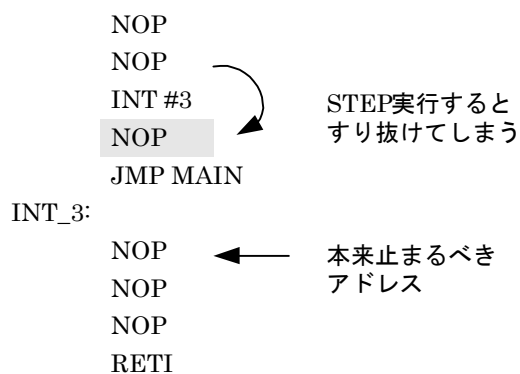
<例>



#### ②INT 命令

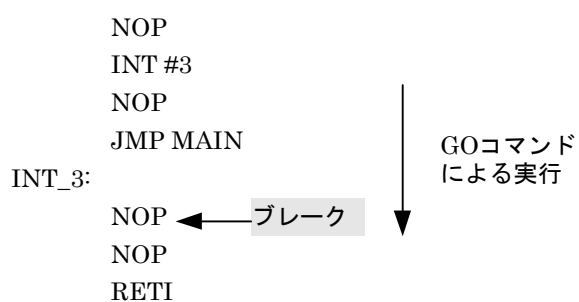
INT 命令から、INT 命令内部処理を連続してSTEP 実行はできません。

<例>



INT 命令を用いたプログラムのデバッグは、INT 命令内部処理にソフトウェアブレイクを設定し、GO コマンドと共に使用して下さい。

<例>



## 7 保証とサポート

### 7.1 保証について

製品の出荷には万全を期しておりますが、欠品、破損、初期不良などがありましたら弊社までご連絡ください。

お問い合わせ先

#### オークス電子株式会社

〒101-0025

東京都千代田区神田佐久間町3丁目21番地

(第一千代田ビル3F)

TEL 03-3863-1121

FAX 03-3863-1130

### 7.2 技術サポートについて

E-MAILにて技術的なご質問を承っております。なおご質問の内容によっては回答に時間がかかる場合がありますのでご了承ください。又弊社サイトでもQ&Aを掲載しております。

E-MAIL: [oaks32support@oaks-ele.com](mailto:oaks32support@oaks-ele.com)

URL : <http://www.oaks-ele.com>

#### <OAKS32シリーズにおける技術的なご質問の制限>

OAKS32シリーズにつきましては、CPUの周辺機能が多岐にわたる為、現在サポートの範囲を限定させて頂いております。

以下の機能につきましてはサポート範囲外となります。

#### 1、CAN全般

#### 2、インテリジェントI/O

##### ・グループ0、1

時間計測機能と波形生成機能のみサポート

通信機能はサポート対象外

##### ・グループ2、3

全機能サポート対象外

#### 3、リングオシレータと発振停止検出回路全般

#### 4、システムクロックPLL全般

#### 5、DMAC2、DRAMC全般

# 付録

部品表  
寸法図

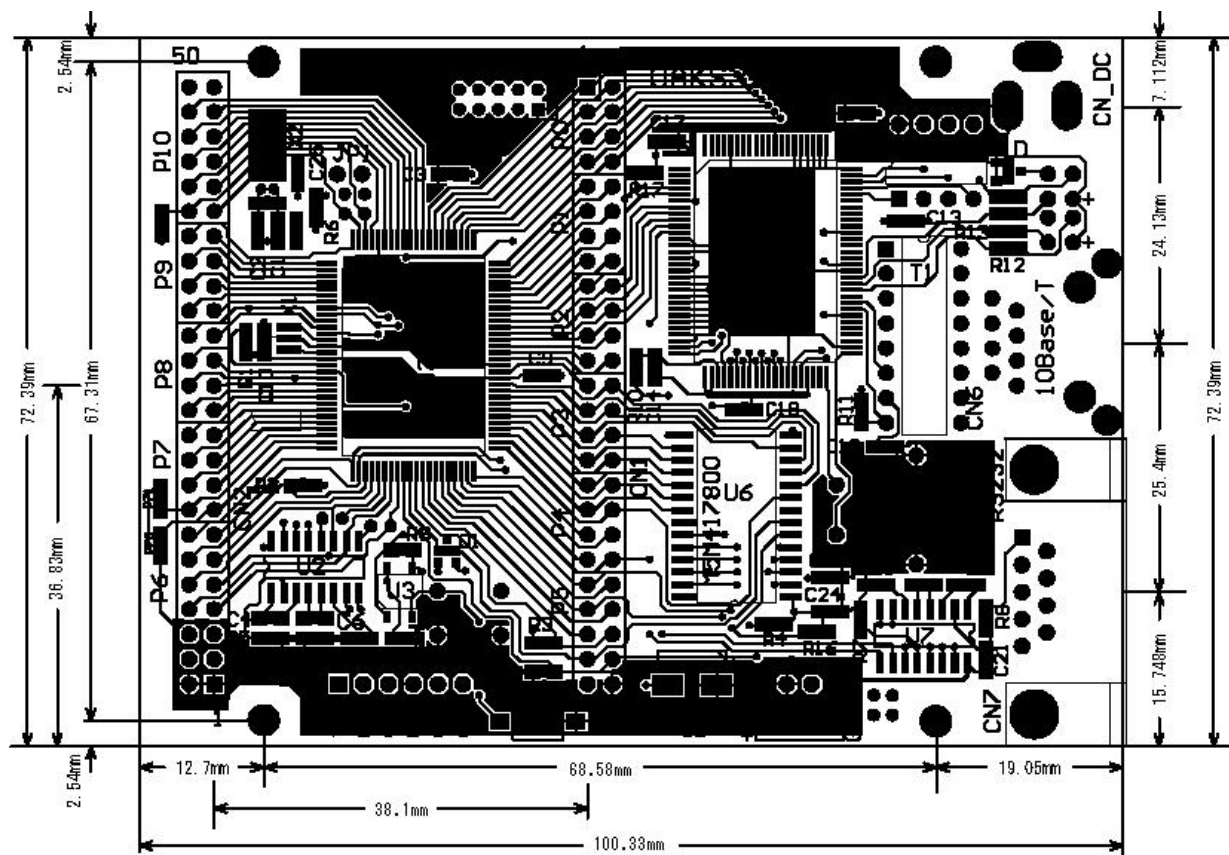
## 部品表

部品番号	パラメータ	部品名	メーカー	部品サイズ	極性	備考
C1,2	GRM1882C1H330JA	33pF セラミック C	Murata	1608		
C3	GRM188B11H105	1uF セラミック C	Murata	1608		
C4~7	GRM188B11H104	0.1uF セラミック C	Murata	1608		
C8~11	GRM188B11H105	1uF セラミック C	Murata	1608		
C12,13	GRM188F11H103ZA	0.01uF セラミック C	Murata	1608		
C14~19	GRM188B11H105	1uF セラミック C	Murata	1608		
C20~23	GRM188B11H104	0.1uF セラミック C	Murata	1608		
C24,25	GRM188B11H105	1uF セラミック C	Murata	1608		
CE	VC-BS 47uF16V	47uF16V ケミコン	TOSHIN		極性あり	
CN1,2	50PinDIP	50PinDIP ヘッダ		50P2.54W		
CN3	DF11-10DP-2DS	2mm ピッチ 10ピンヘッダ		10P2W	極性あり	
CN4	6PinSIP-L	6ピン SIP ヘッダ		6P2.54		
CN5	XA2P	2ピンヘッダ	JST	2P2.54	極性あり	
CN6	RJ45CON	RJ45 コネクタ		RJ45	極性あり	
CN7	RS232	RS232C コネクタ メス		DB9FL	極性あり	
CN8	HEC0470-01-630	DC 電源コネクタ	HOSHIDEN	DCJACK	極性あり	
D1	EBR3378S	赤 3mmΦLED	スタンレー	LED3M	極性あり	POWER
D2	EMBG3378S	緑 3mmΦLED	スタンレー	LED3M	極性あり	LINK
D3	EMAY3378S	黄 3mmΦLED	スタンレー	LED3M	極性あり	RX
D4	EMAY3378S	黄 3mmΦLED	スタンレー	LED3M	極性あり	TX
BLAKET	LW-1	LED2段ブラケット	マックエイト			
BLAKET	LW-1	LED2段ブラケット	マックエイト			
Q1	DTC123YKA	NPN デジトラ	ROHM	SC59	極性あり	
R1~7, 9,10	RK73K1JTD103J	10Kohm 抵抗		1608		

部品番号	パラメータ	部品名	メーカー	部品サイズ	極性	備考
R11	RK73K1JTD201J	200ohm 抵抗		1608		
R8,12~15	RK73K1JTD102J	1Kohm 抵抗		1608		
R16,17	RK73K1JTD103J	10Kohm 抵抗		1608		
U1	M30833FJ	ワンチップマイコン		QFP100	極性あり	モニタプログラム、MR308、TCP308 書き込み済み
U2	SP232ACN	RS232 レベル コンバータ	Sipex	SOP16	極性あり	COM0 用
U3	PST592CM	リセット IC	Mitsumi		極性あり	
U4	RTL8019AS	EtherNetInterface	Realtek	QFP100	極性あり	
U5	93C46	EEPROM	ISSI	DIP8	極性あり	Mac アドレス 書き込み済み
U6	HM5117800TT	16MDRAM	Hitachi		極性あり	
U7	SP232ACN	RS232 レベル コンバータ	Sipex	SOP16	極性あり	COM1 用
T1	LP-164C	10BaseT パルス トランス	LANKOM	DIP16	極性あり	
SW1	SKQDAAE010	プッシュスイッチ	ALPS	SMT-SW		
X1	CSTCG30M0V51-B0	30MHz セラミック 振動子	Murata	CERA-SMT		
X2	32.768KHz	時計用発信子	EPSON	VT100		
X3	20MHz	20MHz クリスタル		HC49U		
JP	Wire	ジャンパーワイヤ				
JP	Wire	ジャンパーワイヤ				
X3-Wire	Wire	水晶固定ワイヤ				

※ CPU以外の部品などについては相当品となる場合があります。

寸法图



## 改訂記録

Version	発行日	ページ	改訂内容
1.00	2003. 8. 1	-	初版発行
2.00	2003. 10. 1	-	三菱電機からルネサステクノロジへ社名変更に伴う修正
	2004. 8. 1	102	R8を10K ohmから1K ohmに修正

OAKS32-LANBoardユーザーズマニュアル Rev 2.00

2003年10月発行

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。