OAKS16でTMをお使いになる方のために

目　次

はじめに

このマニュアルは TM をはじめてお使いになられる方のためのサポートテキストです。基本的には TM V.3.11 のユーザーズマニュアルに準じておりますが、説明の順番、補足説明など変更している部分もあります。OAKS16 をお使いいただく上で、最低限必要と思われる部分だけを抜粋してあります。さらに、詳しい内容をお知りになりたい方は、TM V.3.11 のユーザーズマニュアルを参照してください。

1.ＴＭとは

ＴＭ（ツール・マネージャー）は、コンパイラ／アセンブラ／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、ソフトウエアの開発効率を改善する為のツールです。

ＴＭに、エディタ、コンパイラ、デバッガ、フラッシュＲＯＭ書き込みソフトなどのツールを登録することにより、ＴＭのツールバーから起動させることができるようになります。さらに、コンパイラの起動を make で行っているので、効率的な開発ができます。

1.1.動作環境

ＴＭの動作を確認しているホストマシン、及びＯＳのバージョンについて以下に示します。

ホスト名	ＯＳバージョン
IBMPC/AT 及び互換機	MicrosoftWindows95
	MicrosoftWindows98
	MicrosoftWindowsNT4.0（最新のサービスパックを推奨）
	MicrosoftWindowsMe
	MicrosoftWindows2000 Professional

TM が動作する為には上記の環境に Microsoft Internet Explorer 4.0 以上がインストールされている必要があります。

推奨するハードウエアは以下の通りです。

メインメモリ	ＯＳが正常に動作する状態を推奨（16Mバイト以上）
空きディスク容量	20Mバイト以上
ＣＲＴ	1024×768 以上を推奨

1.2. ＴＭのインストール

以下にＴＭのインストールの手順を示します。

(1)　ＣＤＲＯＭにある\tools\TM\setup ディレクトリ内の setup.exe を実行します。

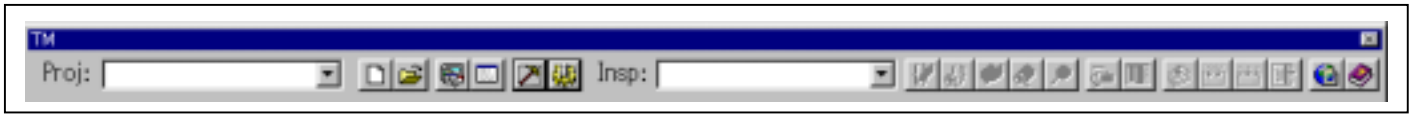
(2)　インストール画面の指示にしたがってインストールします。

2. ＴＭの起動

2.1. ＴＭの起動

スタートメニューから〔スタート〕－〔プログラム〕－〔MITSUBISHI-TOOL〕－〔TM V.3.11〕－〔TM〕をクリックします。

◆　ＴＭが起動すると次のようなプロジェクトバーが表示されます。



2.2. プロジェクトバーボタン

プロジェクトバーの各ボタンの機能を説明します。

ボタン	名称	機能
	プロジェクト名表示ボタン	プロジェクト名の表示
	新規プロジェクト作成ボタン	このボタンをクリックすると、プロジェクトエディタが起動し、新規プロジェクト作成ウイザードがオープンする。
	プロジェクトオープンボタン	プロジェクトファイルをオープンする。
	プロジェクトエディタ起動ボタン	プロジェクトエディタを起動する。既にプロジェクトエディタが起動している場合には,プロジェクトエディタを最前面に表示する。
	エディタ起動ボタン	エディタを起動する。
	ツール登録ボタン	デバッガ、エディタ、アプリケーションを登録する ToolsInformation ダイアログをオープンする。
	カスタマイズボタン	プロジェクトバーの設定を行う Customize ダイアログをオープンする。

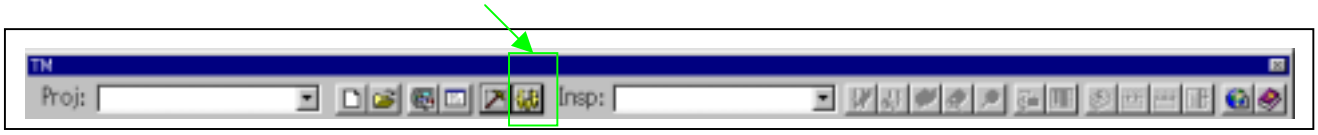
ボタン	名称	機能
	部分ビルドボタン	プロジェクトエディタで選択したアイテムをビルドします。
	ビルドボタン	プロジェクトをビルドする。
	リビルドボタン	プロジェクトをリビルドする。
	デバッグボタン	デバッガを起動する。
	ホームページ表示ボタン	TM のホームページを表示する。
	ヘルプ表示ボタン	本ヘルプを表示する。

（注）“ プロジェクトをビルドする ” とは、コンパイル、アセンブル、リンクという作業によりアブソリュートモジュールファイルを作成することを意味します。ビルドにはＧＮＵの make.exe が使われています。そのため、ビルドにより作成されたファイルとそのファイルのソースになるファイルとのタイムスタンプを比較し、ソースファイルに変更があったと判断された場合（ソースファイルのほうが最近作成されている）のみ、変更された部分に関連するファイルを処理します。
リビルドは、中間生成ファイルと、最終ファイルを削除してからビルドを行うので、全ての作業（コンパイル、アセンブル、リンク）が最初から行われます。

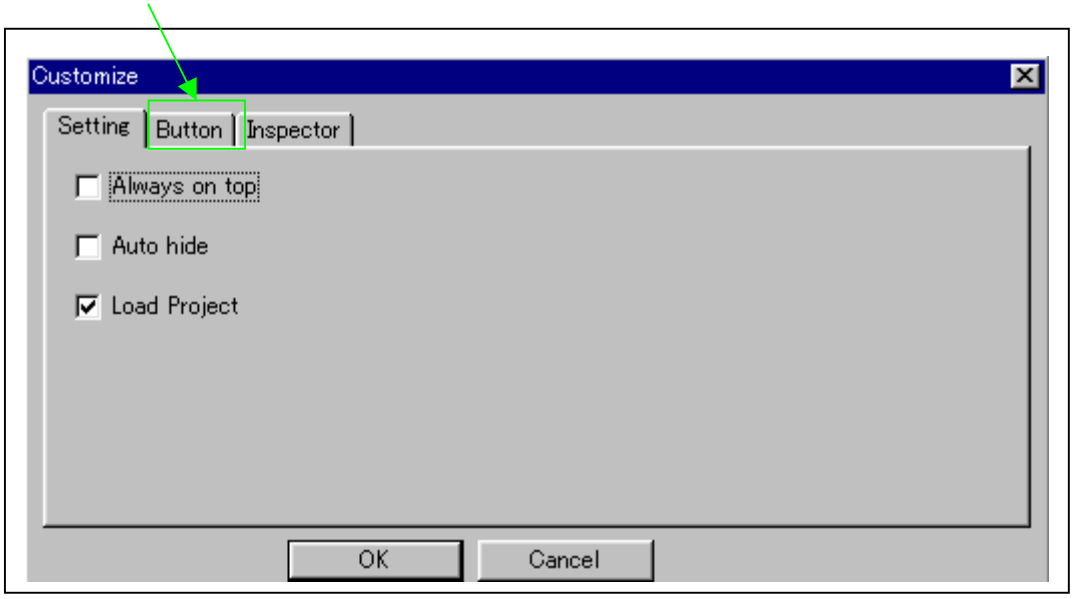
2.3.プロジェクトバーの整理

このプロジェクトバーには OAKS16 版で使用できないボタンが含まれています。
そのままでも問題はありませんが、削除してプロジェクトバーを見やすくしましょう。

(1)　プロジェクトバーの「カスタマイズボタン」をクリックします。

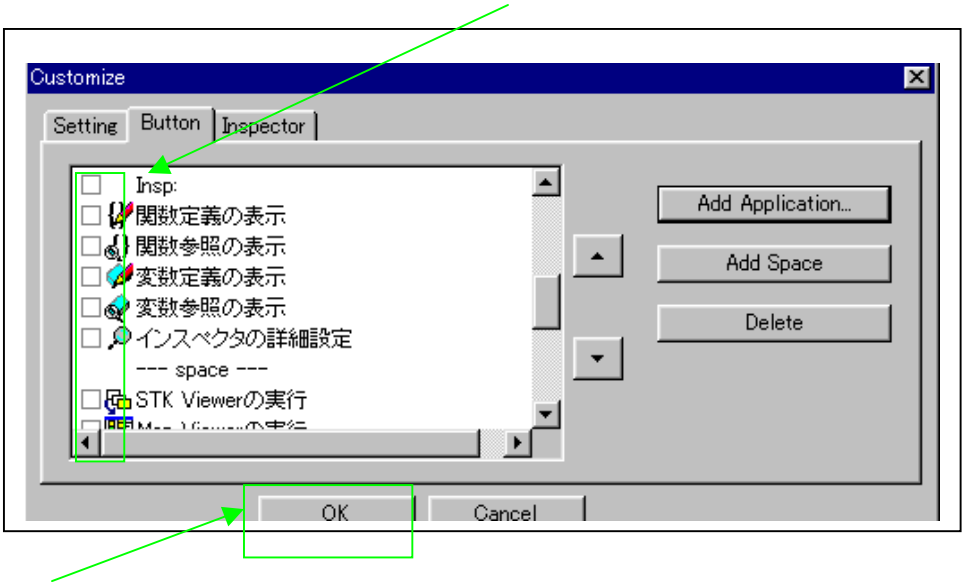


(2)「Button」をクリックします。



(3) 使わないボタンのチェックをはずします。

(Insp から MAP Viewer の実行までチェックをはずします。)



(4)ＯＫをクリックします。

これで、利用できるボタンだけのツールバーができました。

3.ツールの登録

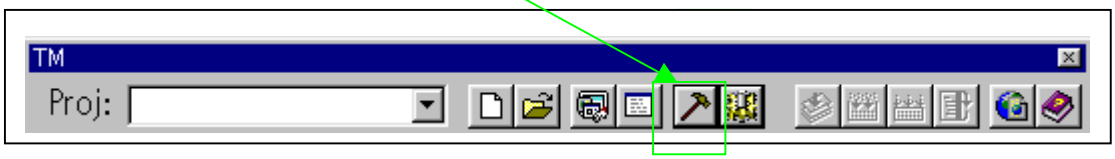
このツールバーに開発で使用するエディタ、デバッガ、フラッシュライタを登録していきます。

これによって、開発のツールはすべてこのツールバーから起動できることになります。（コンパイラは、プロジェクトの作成時に選択しますので、ここでの登録はありません。）登録の前に、NC30WA v4.00、KD30、FlashStart をインストールしてください。

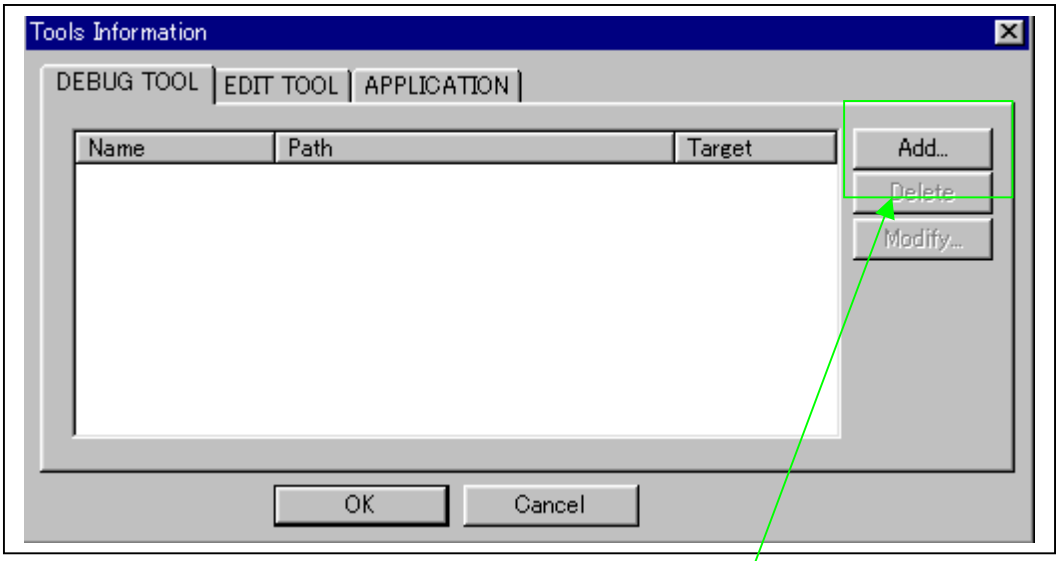
エディタは、OAKS16 に含まれておりませんので、既にお使いいただいているものをそのままお使い下さい。

3.1.デバッガ（KD30）の登録

（ア） ツールバーのツール登録ボタンをクリックします。

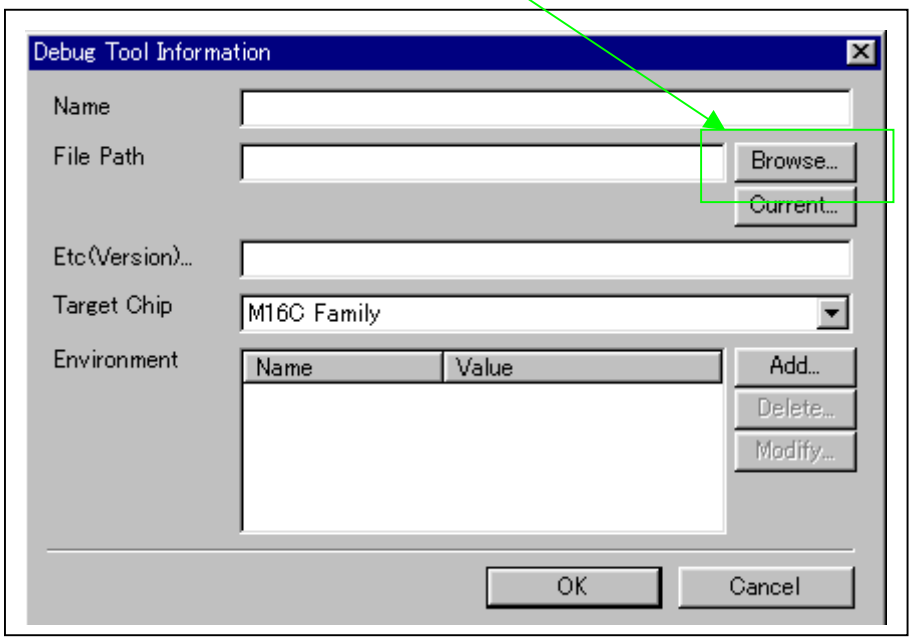


すると、ToolInformation が表示されツールの登録ができる状態になります。



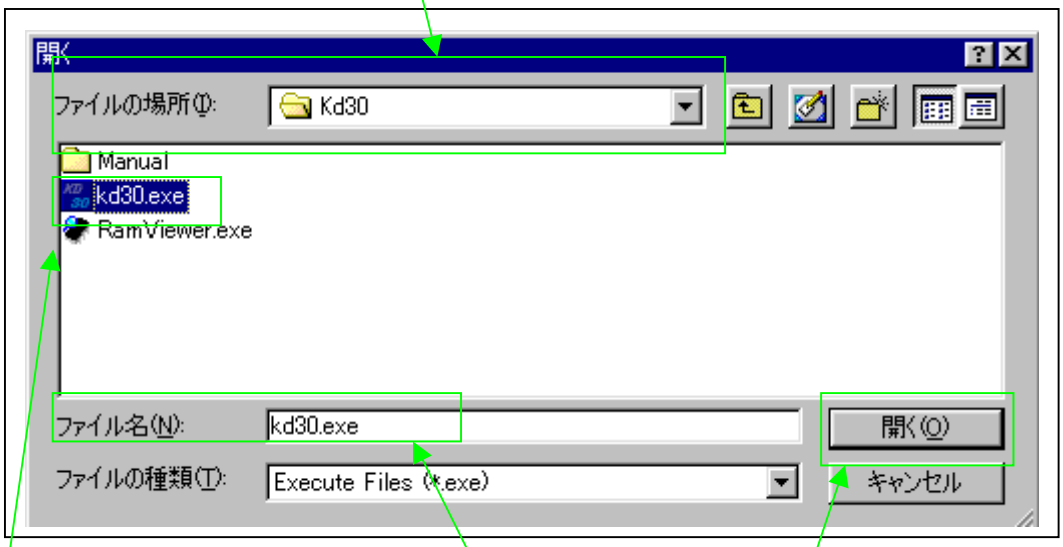
（イ）デバッガ（KD30）を登録する為に DEBUG TOOL 画面で「Add」をクリックします。

（ウ）デバッグツールの設定画面が表示されます。「Browse」をクリックしファイルの場所を指定します。



（エ）KD30 を登録します。

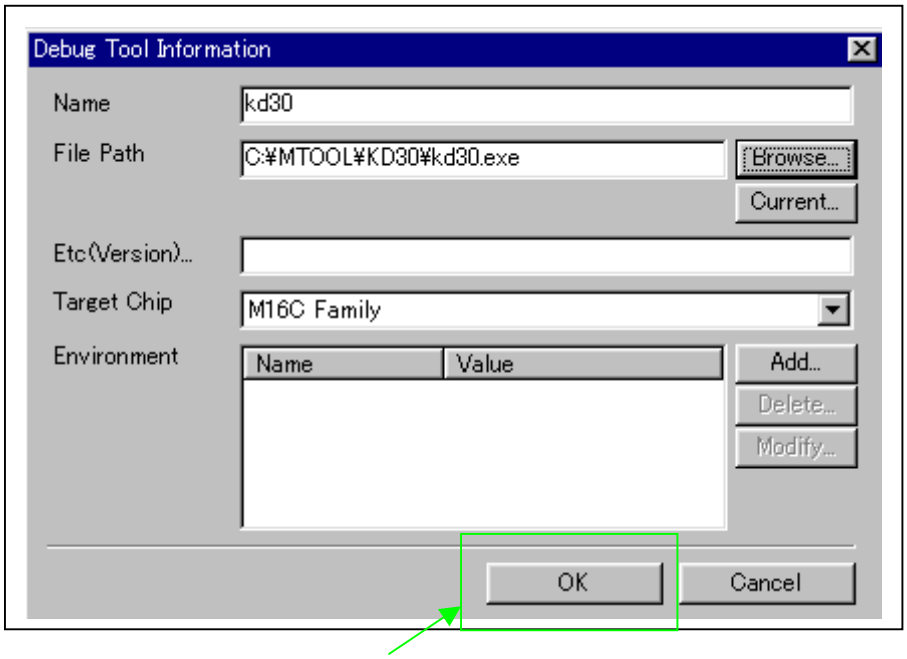
①ＫＤ３０の存在するフォルダを選択します。



②kd30.exe を選択すると、ファイル名に kd30.exe が入ります。

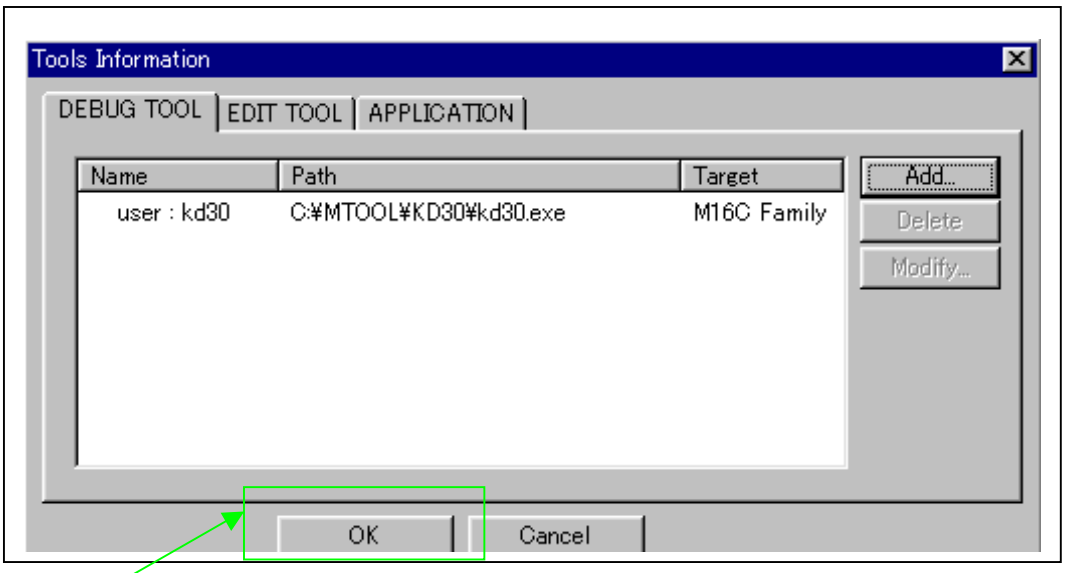
③「開く」をクリックします。

（オ）DebugToolInformation に kd30 が表示されます。



「ＯＫ」をクリックします。

（カ）DEBUGTOOL に KD30 が追加されました。



「ＯＫ」をクリックします。

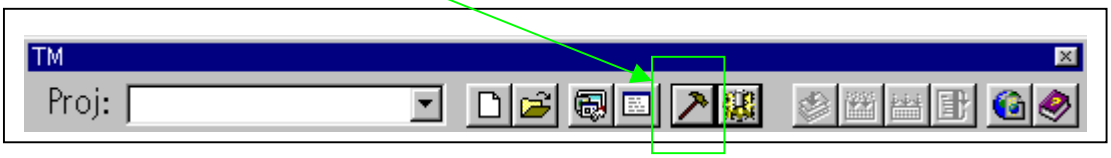
（キ）これでデバッガの設定が終了です。デバッグボタンをクリックすれば KD30 が起動します。

3.2.エディタの登録

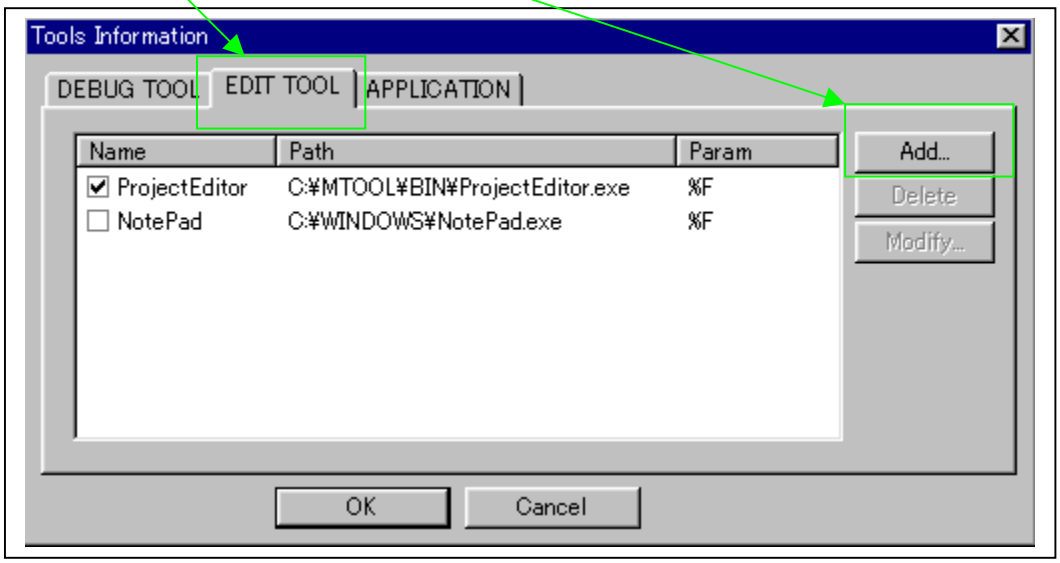
次にエディタの登録をします。

TM では、内部にエディタを持っていません。お好みのエディタを自由に登録することができます。ここでは例としてアンカーシステムズ社の「Peggy」エディタを登録します。

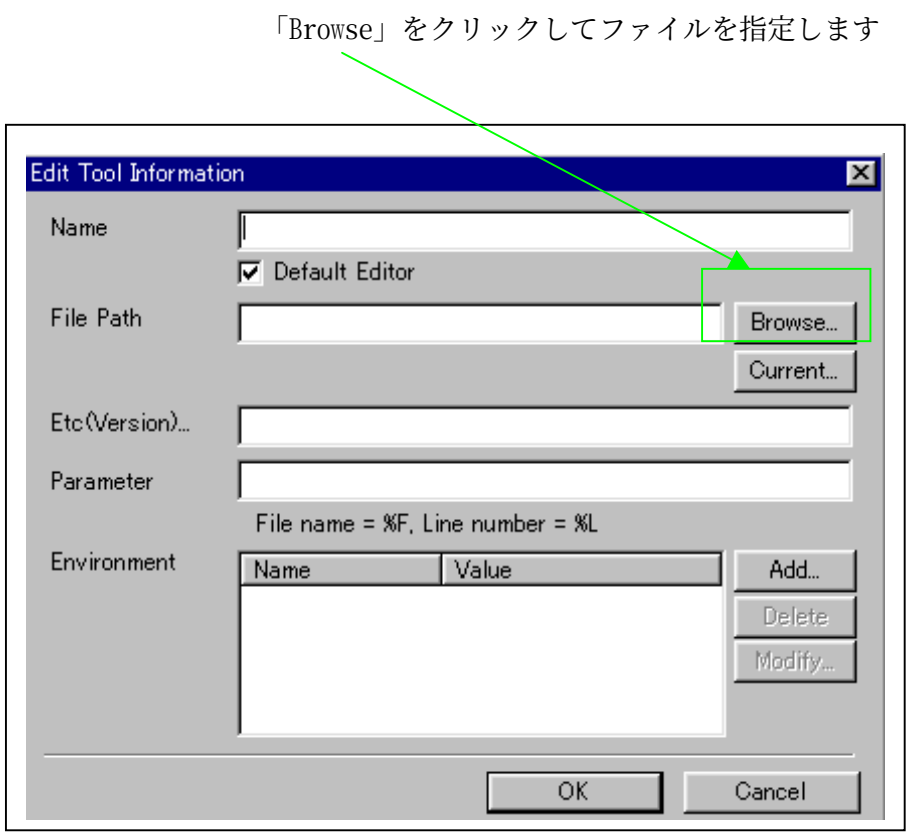
（ア） ツールバーの「ツール登録ボタン」をクリックします。



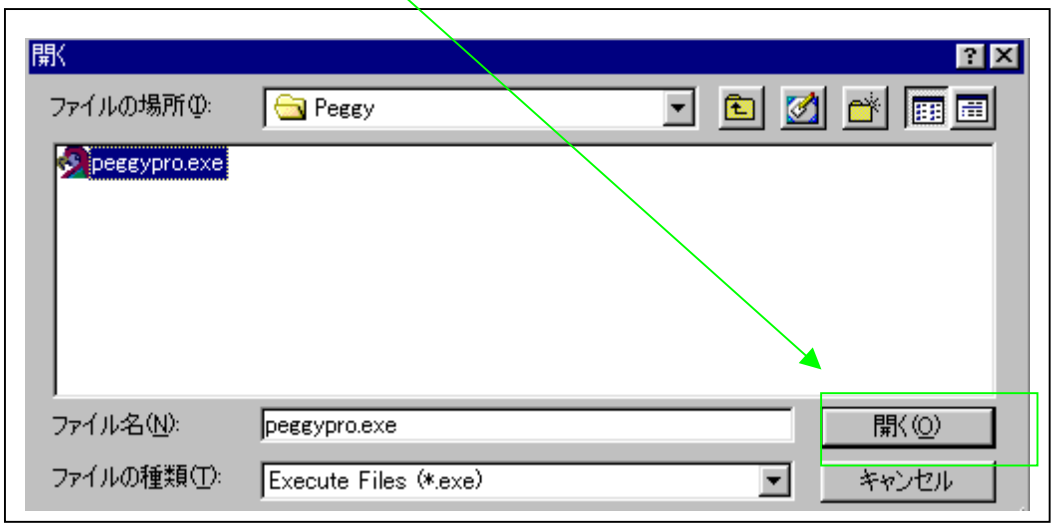
（イ） 「EDITTOOL」を選択し「Add」をクリックします。



（ウ）EditToolInformation の画面が表示されます。



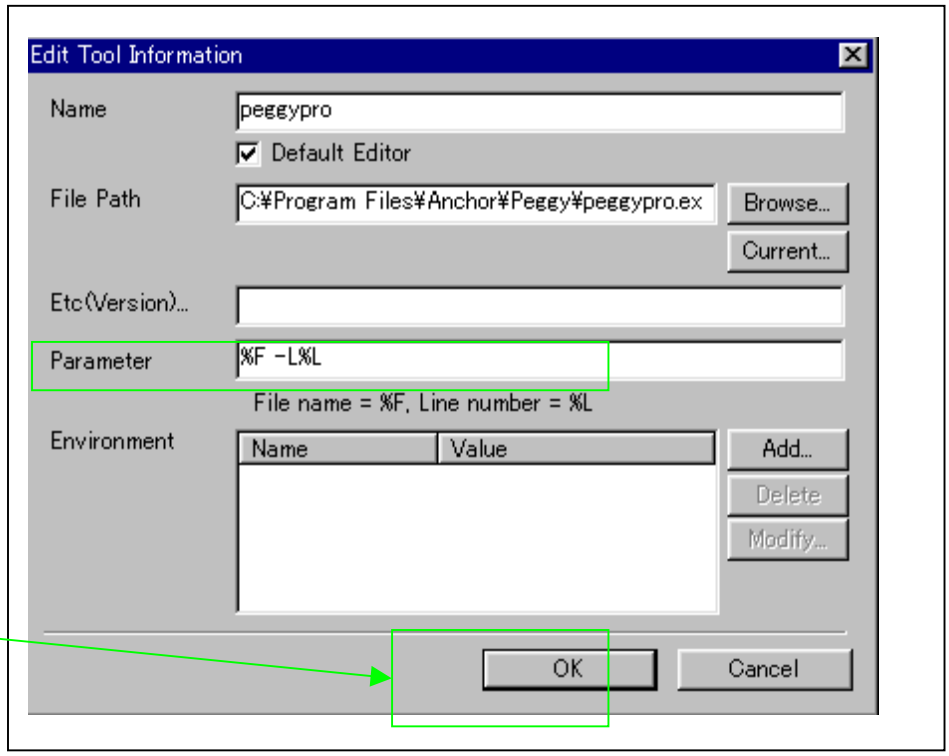
（ウ） Peggypro.exe を指定し「開く」をクリックします。



（エ）EditToolInformation の画面に peggypro が設定されました。ここでパラメータを設定します。これは、ビルド（コンパイル、アセンブル,リンク）を行ったときの ERR 表示から、ERR のあるファイルを開き、ERR 行にカーソルを移動させる為のものです。TM は、起動されるエディタに対し、ファイル名を%F で、ライン行を%L で引き渡します。登録するエディタのＨＥＬＰからファイルを開くためのコマンドと引数の記述を確認し、パラメータの設定をして下さい。

パラメータの設定

［ＯＫ］をクリックします。



以下にエディタの HELP 記述例を引用します。今回この記述が「ＴＭ」にあてはまります。

参考：peggypro　HELP より引用

例】コンパイラメーカーの統合ツールから Peggy でソースファイルを開き、エラータグジャンプしたい場合は、以下のように引数を設定してください。

ファイル名が%F

行番号が%L

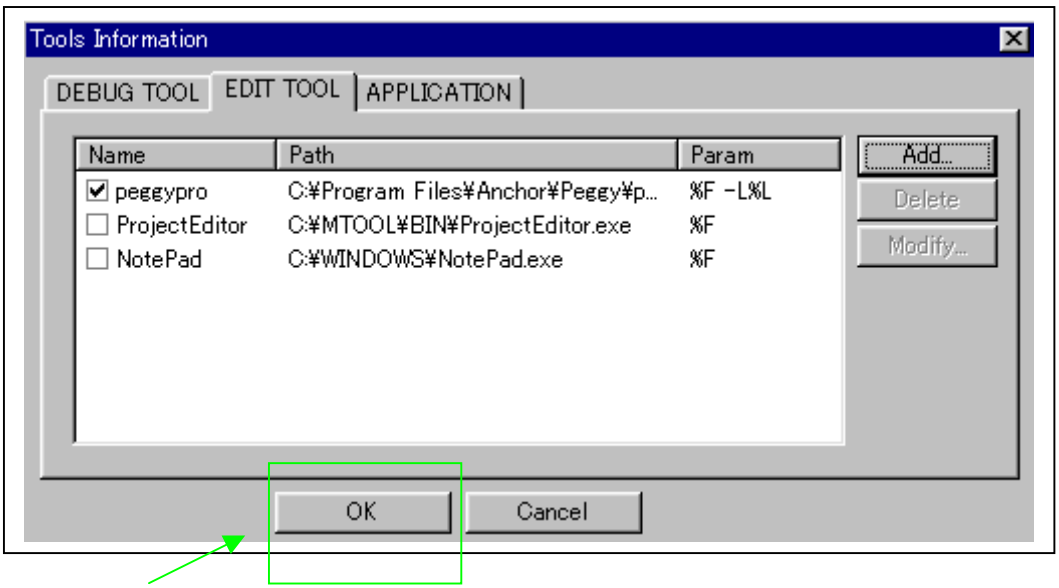
で引き渡される場合。

Peggy %F -L%L

と登録することで、エラータグジャンプができます。

（注意）「peggypro」エディタは、アンカーシステムズ株式会社の製品です。
詳細等はこちらへお問い合わせください。
http://www2.noritz.co.jp/anchor/index.html

（エ） EDHITOOL に peggypro が追加されました。［ＯＫ］をクリックして登録終了です。
ツールバーのエディタボタンをクリックすれば peggypro が開きます。

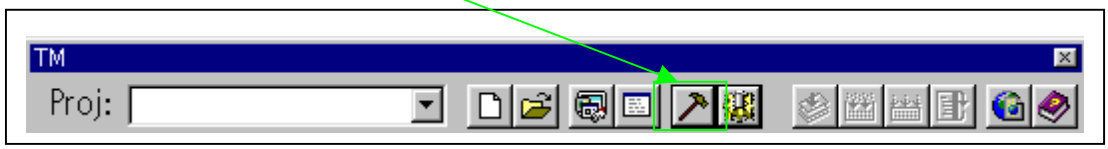


クリックします。

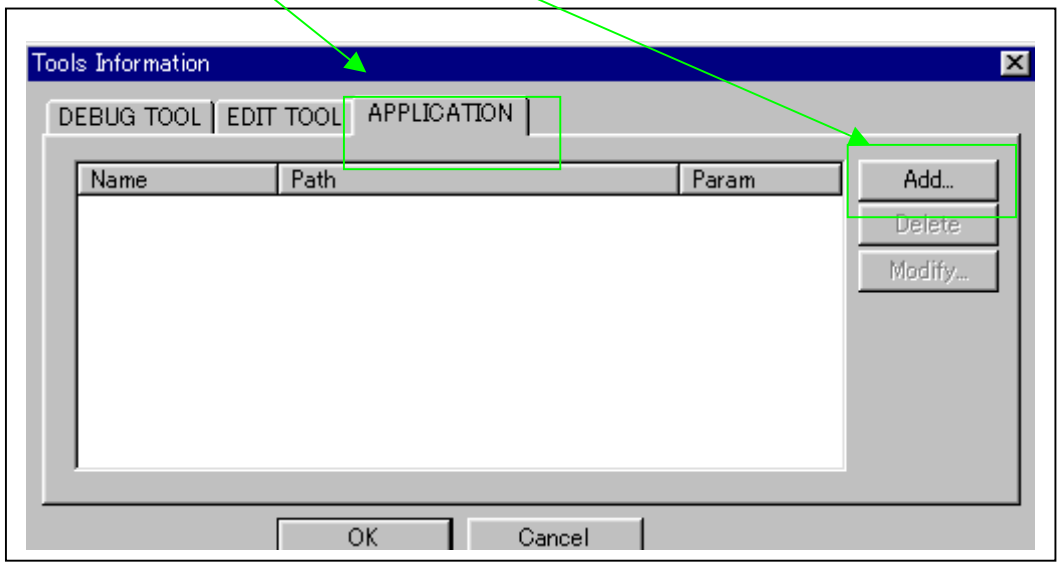
3.3.フラッシュ ROM 書き込みツール（Flashstart）の登録

次に、FlashStart を登録します

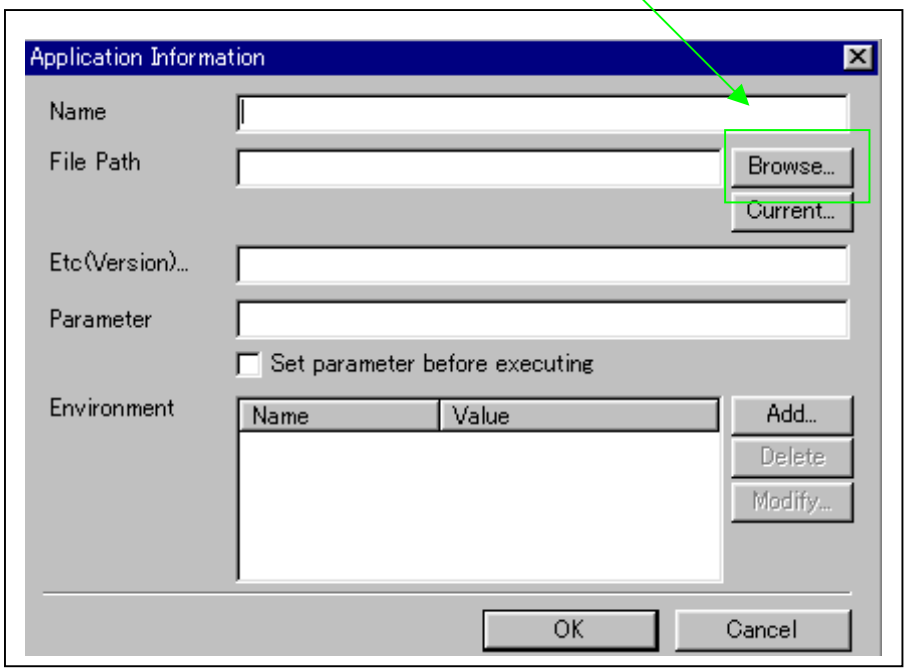
（ア）ツールバーの「ツール登録ボタン」をクリックします。



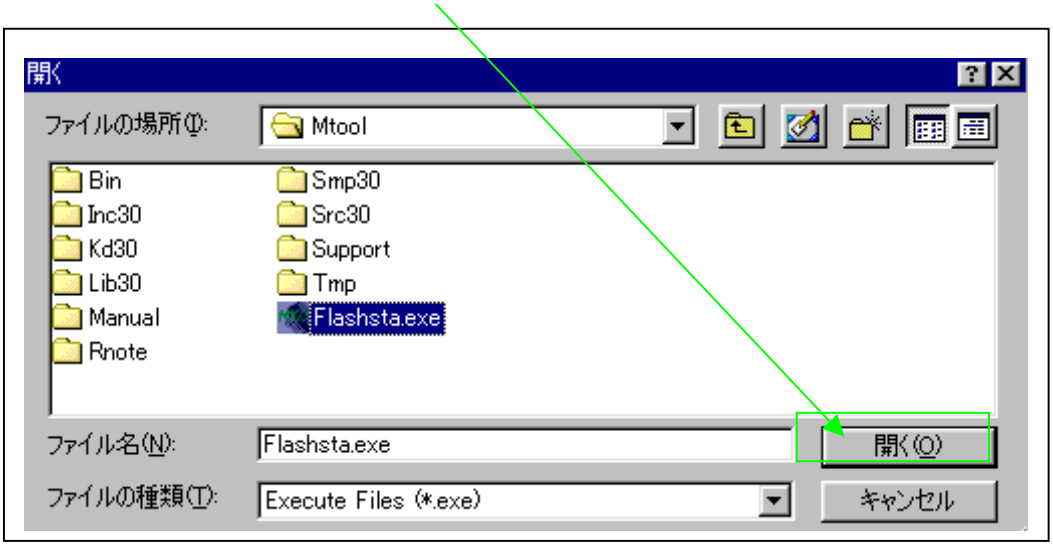
（イ）APPRLICATION を選択し「Add」をクリックします。



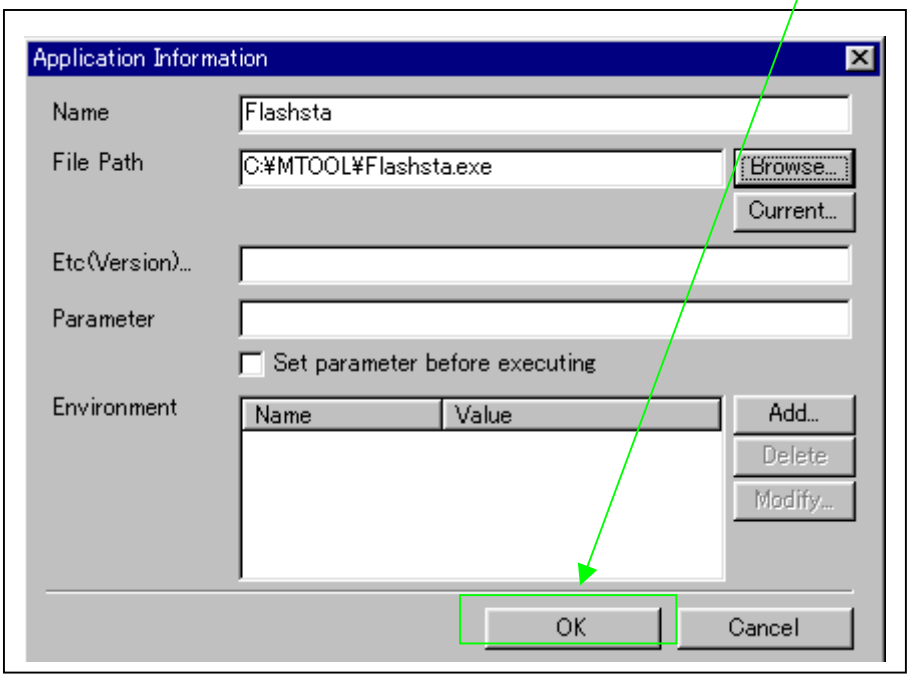
（ウ）ApplicationInformation の画面が表示されました。「Browse」をクリックしてファイルを設定します。



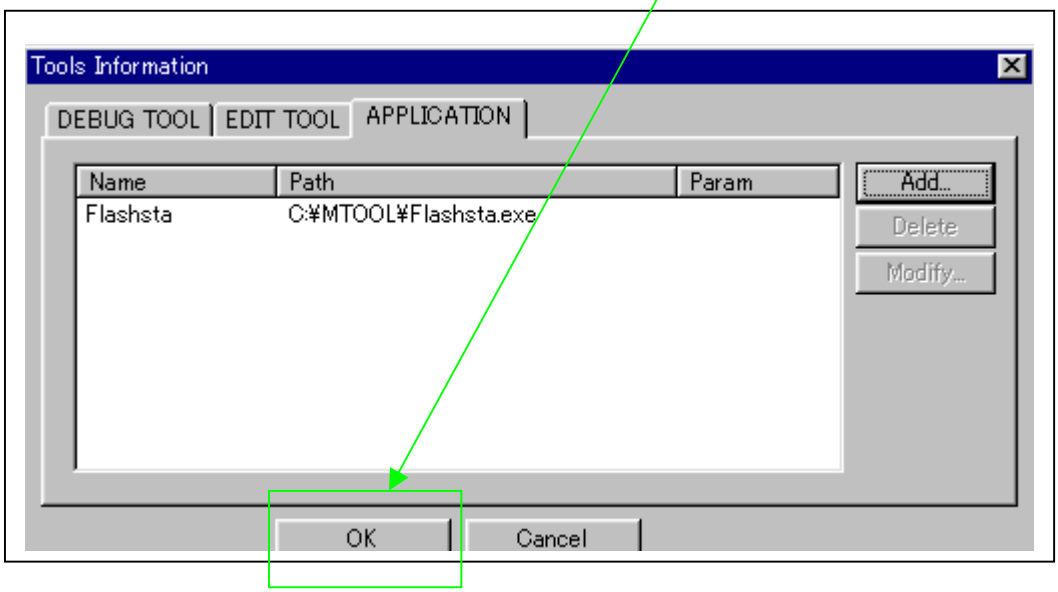
（エ）Flashsta.exe を選択し「開く」をクリックします。



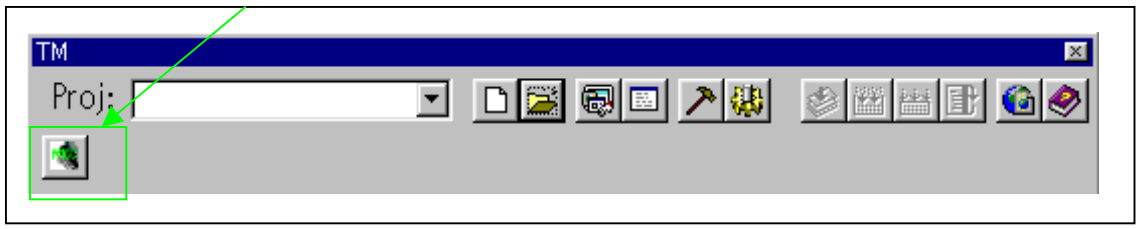
（オ）ApplicationInformation に Flashsta.exe が設定されました。「ＯＫ」をクリックします。



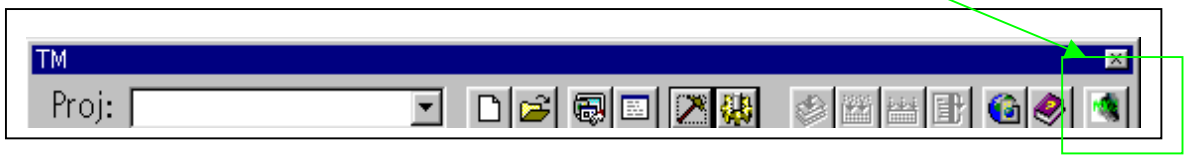
(カ)ToolsInformation に Flashsta.exe が追加されました。「ＯＫ」をクリックして終了です。



（キ）ツールバーに FlashStart のボタンが追加されました。



（ク）ツールバーを横に伸ばします。これで終了です。FlashStart ボタンをクリックすれば FlashStart が起動します。



4.make とは

TM は、プログラムファイル作成の為のコマンド起動を、「GNUmake」を使って行っています。TM がどのようにプログラム開発をサポートしているのかを理解する為に、make の事を少し知っていただきたいと思います。

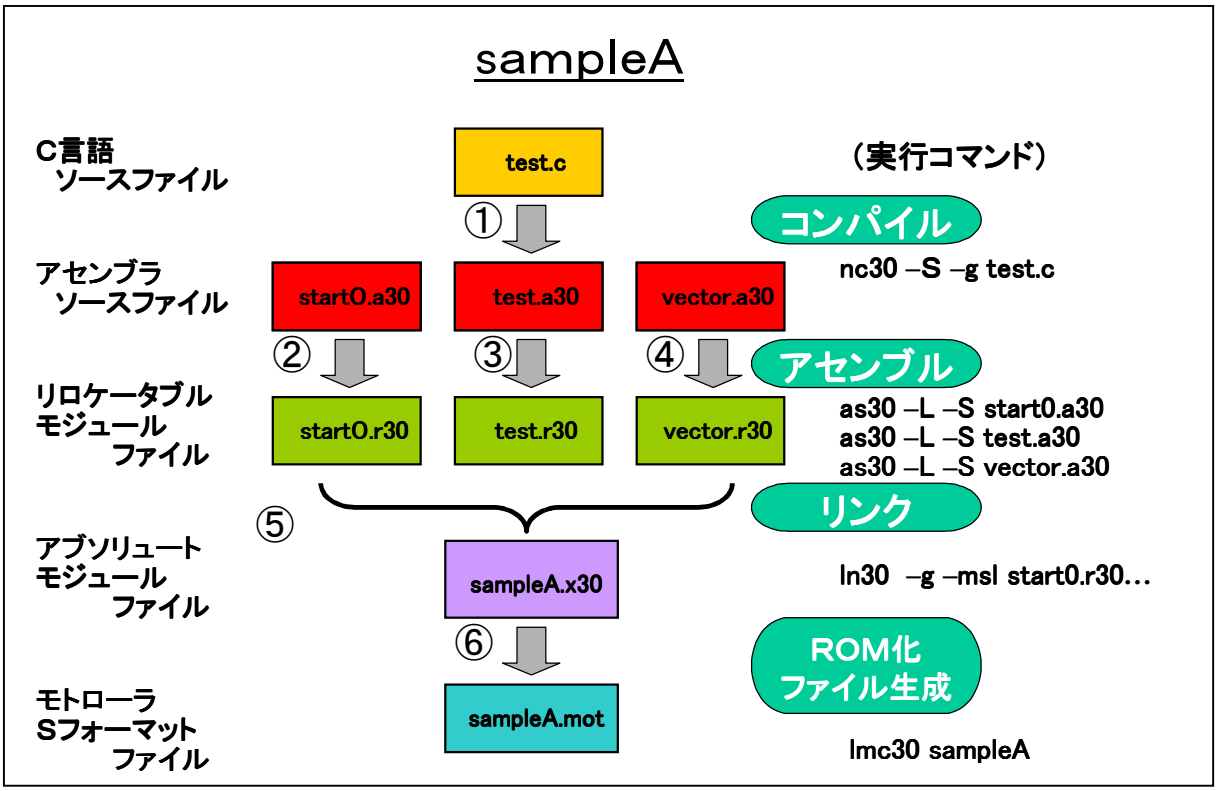
make は、AT&T ベル研究所で UNIX 用に開発されたプログラム保守ツールです。C 言語は UNIX オペレーティングシステムの開発のための言語ですから、C 言語プログラム開発では、多くの場合 make が使われます。

make は、コンパイラに付属されている場合もありますが、今回 TM で使っている make は、GNU プロジェクトで作られたものです。GNU は「GNU ’s Not Unix」（GNU は Unix ではない）の再起頭文字であり「グニュー」と発音されます。「GNUmake」はこの GNU プロジェクトで開発されたフリーソフトウエアで、UNIX の世界では、大変ポピュラーなものです。

この make の簡単な使い方を、sample プログラムを使って説明していきましょう。

4.1.make の考え方

例として、OAKS16 のＣＰＵボード上のＬＥＤ１を点滅させるプログラムを考えてみましょう。



ソースファイルは test.c(C 言語ソースファイル)、start0.a30（アセンブラ言語ソースファイル)、vector.a30（アセンブラ言語ソースファイル）の 3 つです。

① test.c からはコンパイラ NC30 により、test.a30 というアセンブラソースファイルが生成されます。

② start0.a30 からはアセンブラ AS30 により、start0.r30 が生成されます。

③ test.a30 からはアセンブラ AS30 により、test.r30 が生成されます。

④ vector.a30 からはアセンブラ AS30 により、vector.r30 が生成されます。

⑤ ②③④で作成されたリロケータブルモジュールファイルをリンクして、sampleA.x30 というアブソリュートモジュールファイルを作成します。（デバッガ KD30 用）

⑥ ⑤で作成された sampleA.x30 から sampleA.mot というモトローラ S フォーマットのファイルを生成します。（FlashStart 用）

ここまでが、開発の流れです。

ここで、全ての作業が終了してから、①のソースファイルだけに変更が発生したとします。ここで②と④は変更していないのでそのままにし、①、③、⑤、⑥を実行した方が目的のファイルを生成するための時間が短くてすみます。これを実現するのが make です。

make では makefile にこの①から⑥のような依存関係と実行コマンドを記述します。そして、それぞれのファイルの作成された日時に注目し、ソースファイルのほうに変更があった場合だけ、そのファイルに関連するコマンドを実行し、新しいオブジェクトファイルを作成します。変更されていないソースファイルに関しては、コンパイル、アセンブル等を行わず、以前に作成されたリロケータブルファイルをそのまま使います。この作業を行う実行ファイルが、make.exe です。

4.2.make の記述方法

次に簡単に make の記述について説明します。

make の記述方法

ターゲット：ソース１　ソース２　...　ソースｎ
　　コマンド

ここでコマンド行の先頭には一つ以上のタブが必要です。

この記述の意味するところは、

「ターゲットは、ソース１からソースｎまでのファイルによって生成される。その作成手順はコマンドが示している。」

ということです。

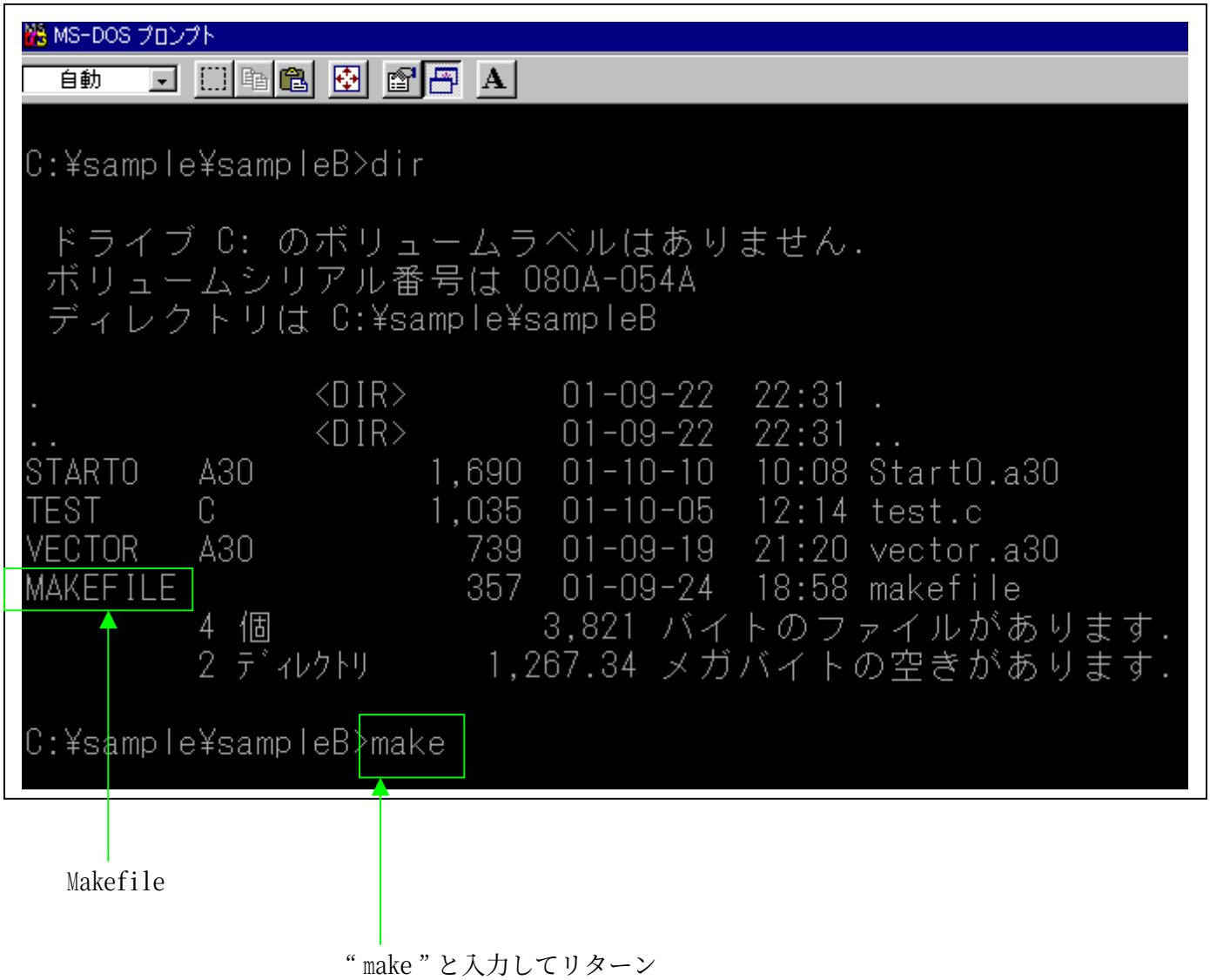
make はこれを見て、まずターゲットという名前のファイルが存在するかどうかをチェックします。もし、存在しなければ、コマンドを実行します。また、ソースファイルのどれかがターゲットより新しければコマンドを実行します。そうでなければ、何もしません。ソースが、どこか他の行でターゲットに指定されている場合は、その行を先に実行します。

この make を記述したファイルを makefile と呼びます。ファイル名は拡張子無しの“makefile”です。

この記述にもとづいて sampleA の makefile を作成すると次のようになります。

```
sampleA.mot:sampleA.x30
        lmc30 -ID#0 sampleA.x30
sampleA.x30 : start0.r30 test.r30 vector.r30
        ln30 -G -MSL start0.r30 test.r30 vector.r30 -O sampleA -ORDER program=0e0000
start0.r30 : start0.a30
        as30 -L -S start0.a30
test.r30 : test.a30
        as30 -L -S test.a30
vector.r30 : vector.a30
        as30 -L -S vector.a30
test.a30 : test.c
        nc30 -S -g test.c
```

この make を試すことができます。OAKS16CDROM の\oaks16\sample 内に sampleB というフォルダがあります。ここに、3 つのソースファイルと、makefile が入っています。オブジェクトファイル名が sampleB になっていますがソースファイルは変わりません。ハードディスクの任意のフォルダにコピーしてください。(できれば、ルートディレクトリに\sample をディレクトリごとコピーしていただくと、この後の章での確認もしやすくなります。) DOS 窓を開いて、sampleBのディレクトリに入り“make”とコマンド入力してください。TMをインストールしたときに\mtool\bin にコピーされた make.exe が起動して、make が実行されます。

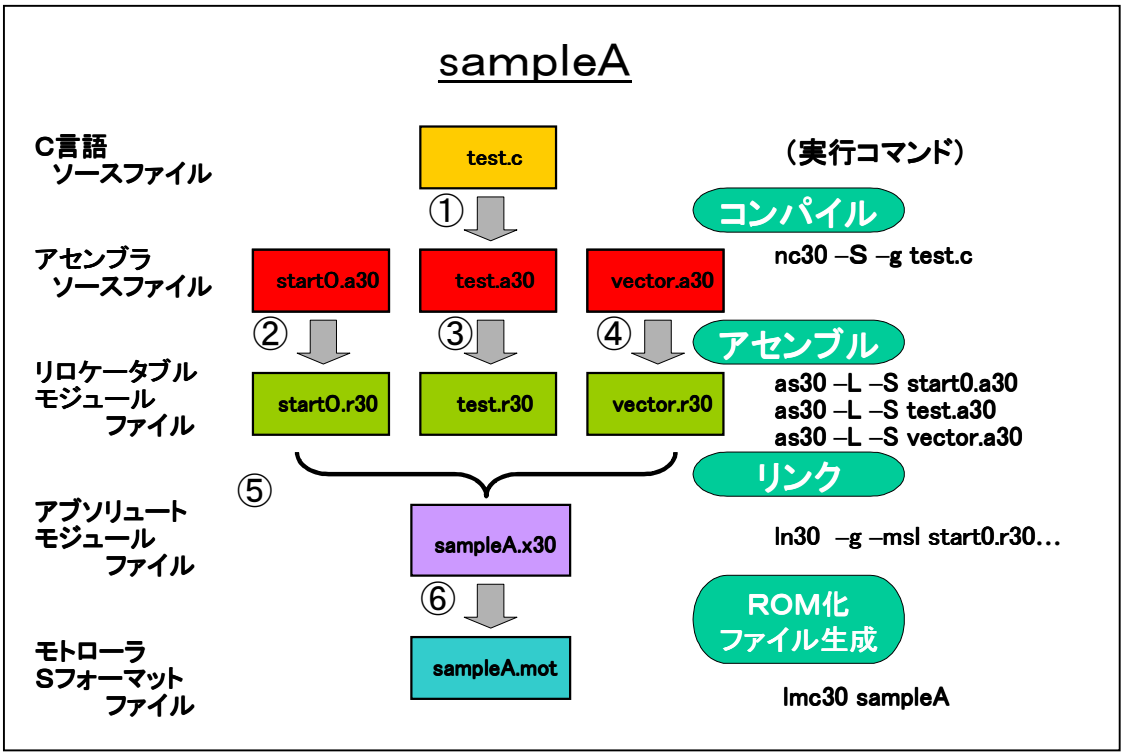


5 プロジェクトの作成

5.1. プロジェクトとは

ＴＭでは、プログラムの開発単位をプロジェクトとして管理します。プロジェクトの作成では、プロジェクトエディタを使用し、プログラムを作成する為のソースファイルを登録します。この関係から、ＴＭではメイクファイルを作成し、gnumake を使用して、オブジェクトファイルを作成します。

先ほどの make の項目で使用した sample で考えてみましょう。



ソースファイルは、次の 3 つです。

・test.c（ソフトウエイトを使って LED を点滅する為の C 言語ソースファイル）

・start0.a30（M16C の初期設定の為のアセンブラソースファイル）

・vector.a30（ベクターアドレス定義のアセンブラソースファイル）

オブジェクトファイルはデバッガ（KD30）でロードする為のアブソリュートモジュールファイル（拡張子ｘ30）です。そして、フラッシュＲＯＭに書き込むためにはモトローラＳフォーマットのファイル（拡張子 mot）が必要です。この 2 つのファイルを作成する為にプロジェクトを組みます。ＴＭでは、プロジェクト名がオブジェクト名になりますので、プロジェクト名を sampleA、オブジェクトファイルを sampleA.x30、sampleA.mot とすることにします。

プロジェクトはフォルダ単位で作成しますので、今回のプロジェクトの為のフォルダを作成します。

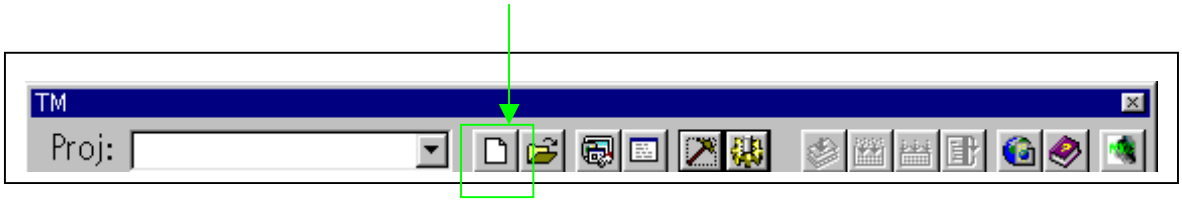
注意：ＴＭには、ファイルについての制限事項があります。

①漢字（２バイト文字）を含むディレクトリ名、ファイル名は使用できない。
②ファイル名に使用するピリオド（.）は一つのみ。
③ネットワークパス名は使用できない。
④「ショートカット」を仕様できない。
⑤空白文字を含むディレクトリ名、ファイル名は使用できない。
例）“ My Documents ”など
⑥“ . . ”表記を用いて二つ以上のディレクトリを指定することができない。
⑦パス指定を含めたファイル名の長さが 128 文字以上になるものは使用できない。

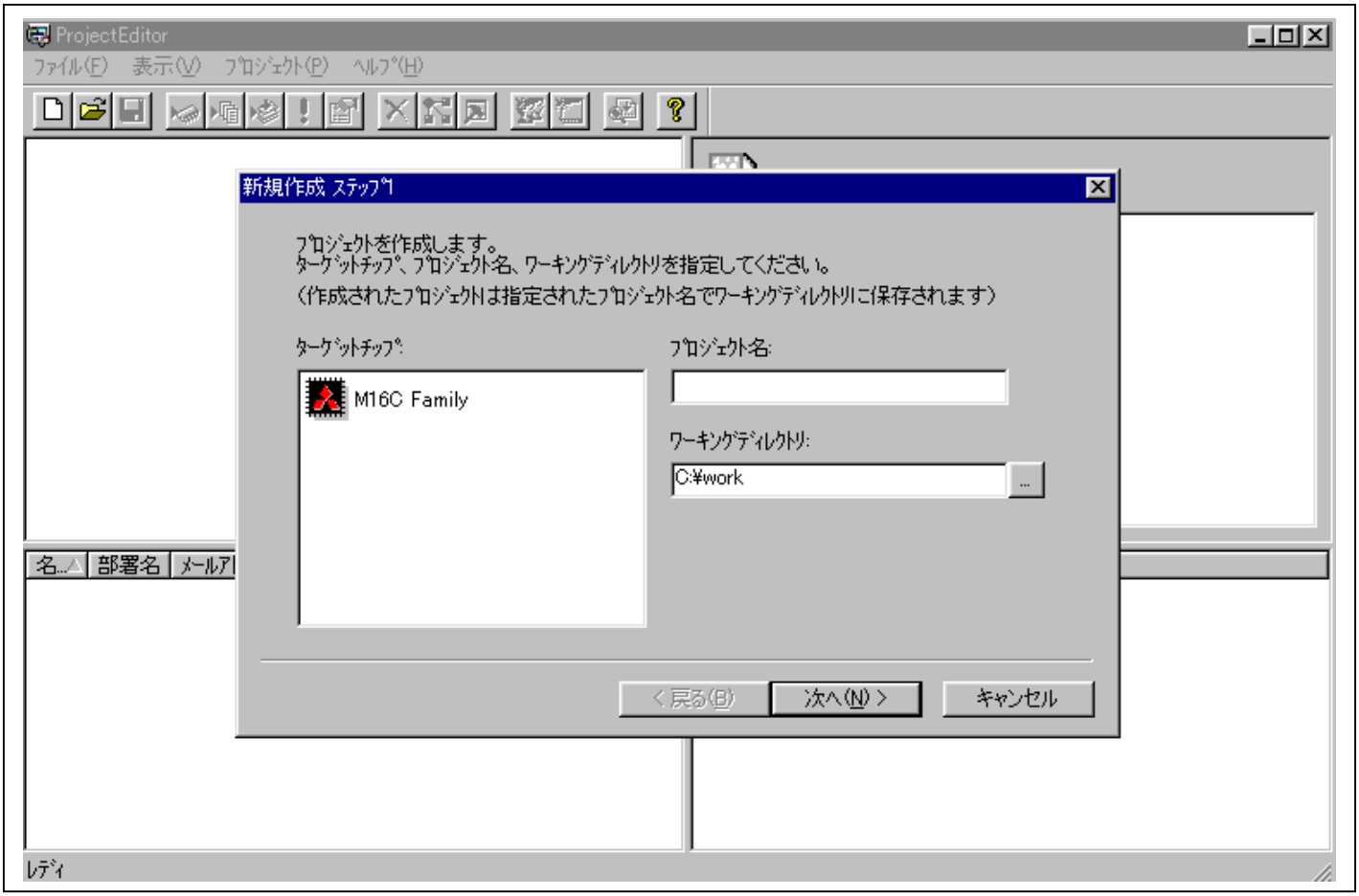
この制限を踏まえて、今回はハードディスクのルートにＣＤＲＯＭの\sample というフォルダをコピーします。その下には\sampleA というフォルダがあります。\sampleA には 3 つのファイル（test.c、start0.a30、vector.a30）があります。（4.make の章で\sample のディレクトリをコピーしてくださった方はそれを使ってください。）

5.2.基本プロジェクトの作成

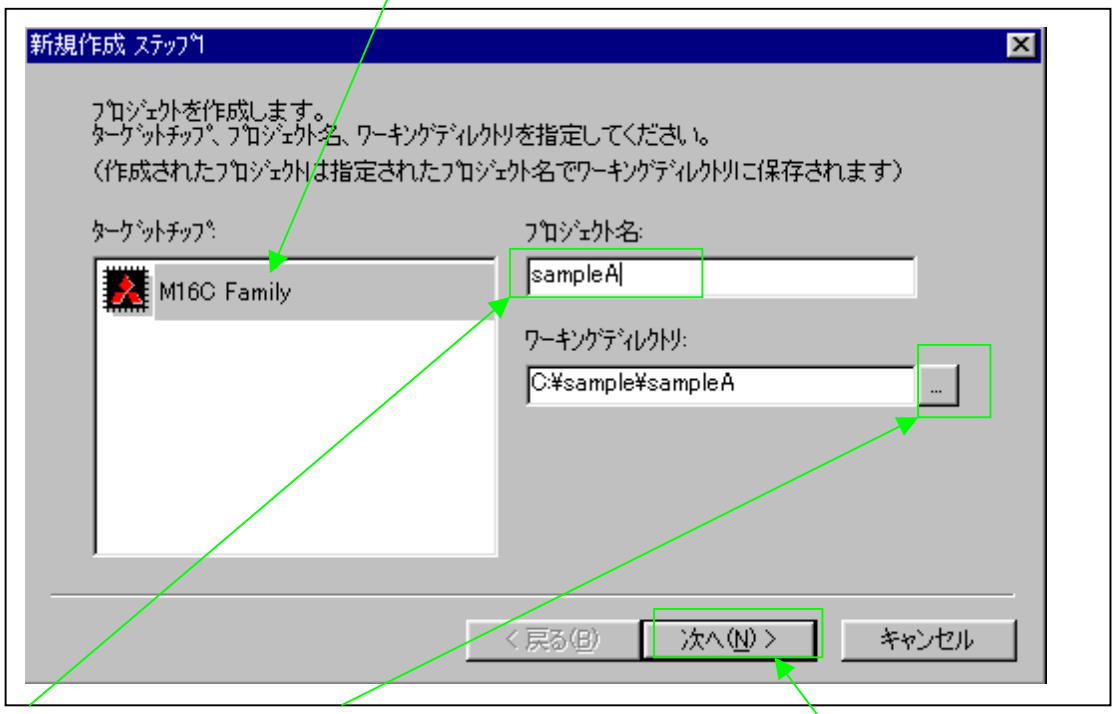
（ア） ツールバーの「新規プロジェクト作成ボタン」をクリックします。



（イ） プロジェクトエディタが表示され、その前面に新規作成ウィザード「新規作成ステップ 1」の入力画面が表示されます。



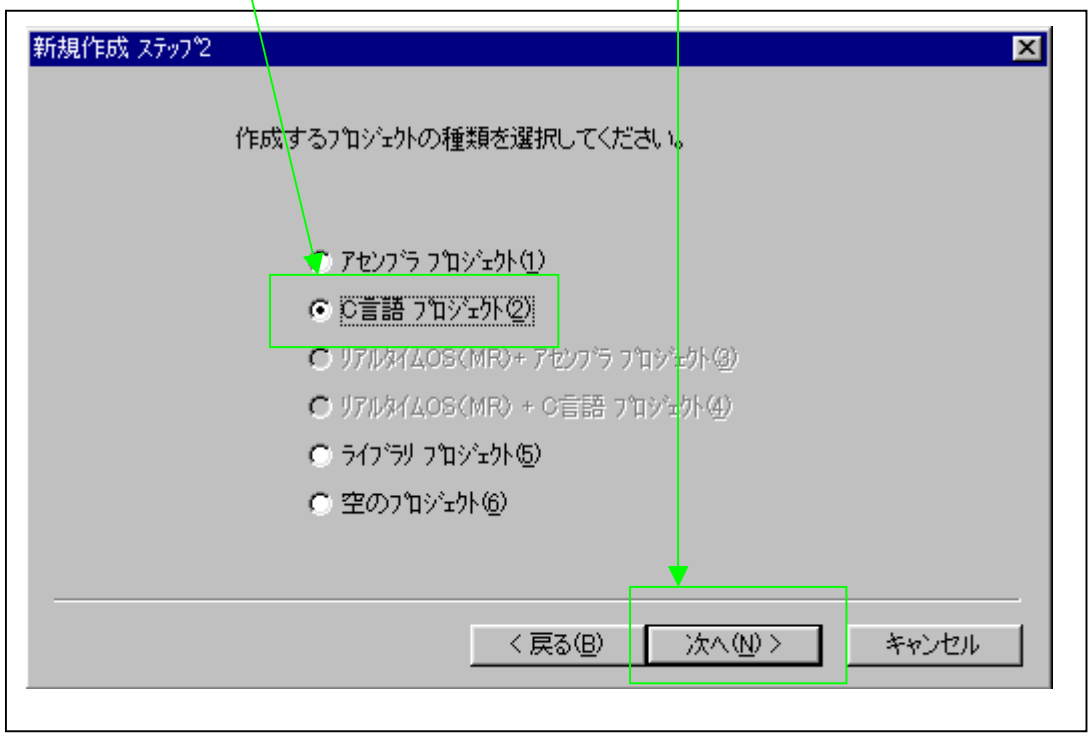
（ウ）ターゲットチップ、プロジェクト名、ワーキングディレクトリを指定します。三菱のチップに対するコンパイラとして NC30WA だけしかインストールしていませんのでこのシステムではターゲットチップとして M16CFamily しか表示されませんが、他のコンパイラをインストールしてあるパソコンでは他にもターゲットチップが表示されます。一つしか表示されていなくても、必ずクリックして、背景がグレーになり選択されるのを確認してください。プロジェクト名、ワーキングディレクトリは「5.1 プロジェクトは」で説明したものです。



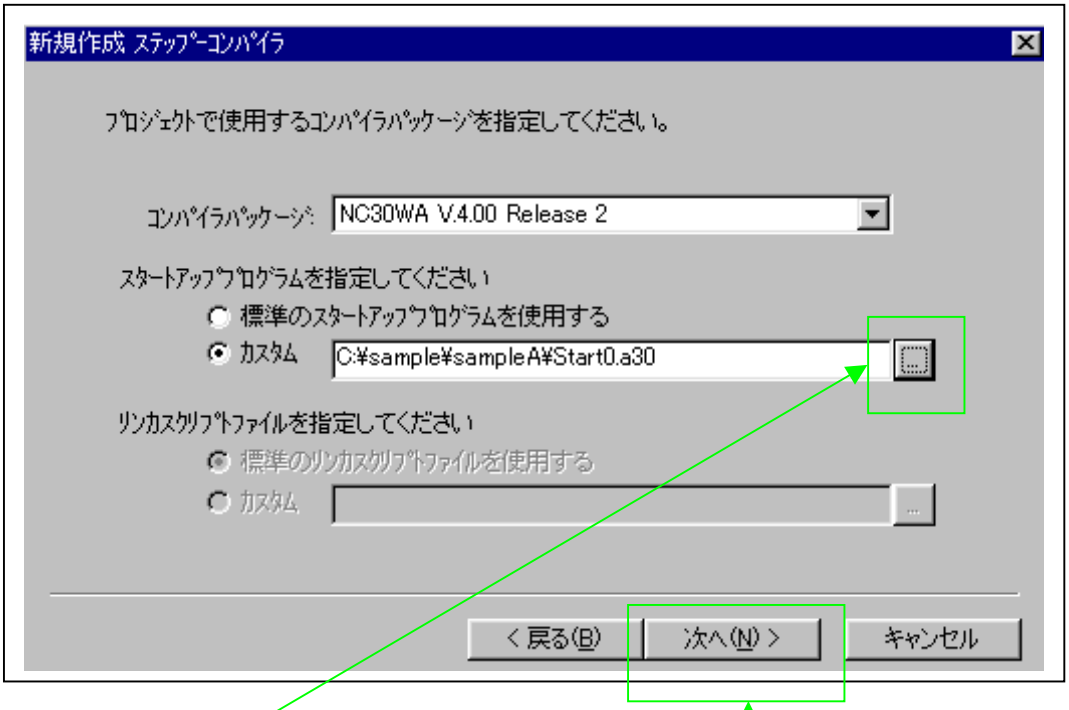
「sampleA」と入力します。このボタンをクリックしてディレクトリを指定します。

ディレクトリが指定できたら「次へ」をクリックします。

（エ）Ｃ言語プロジェクトを選択します。　　「次へ」をクリックします。



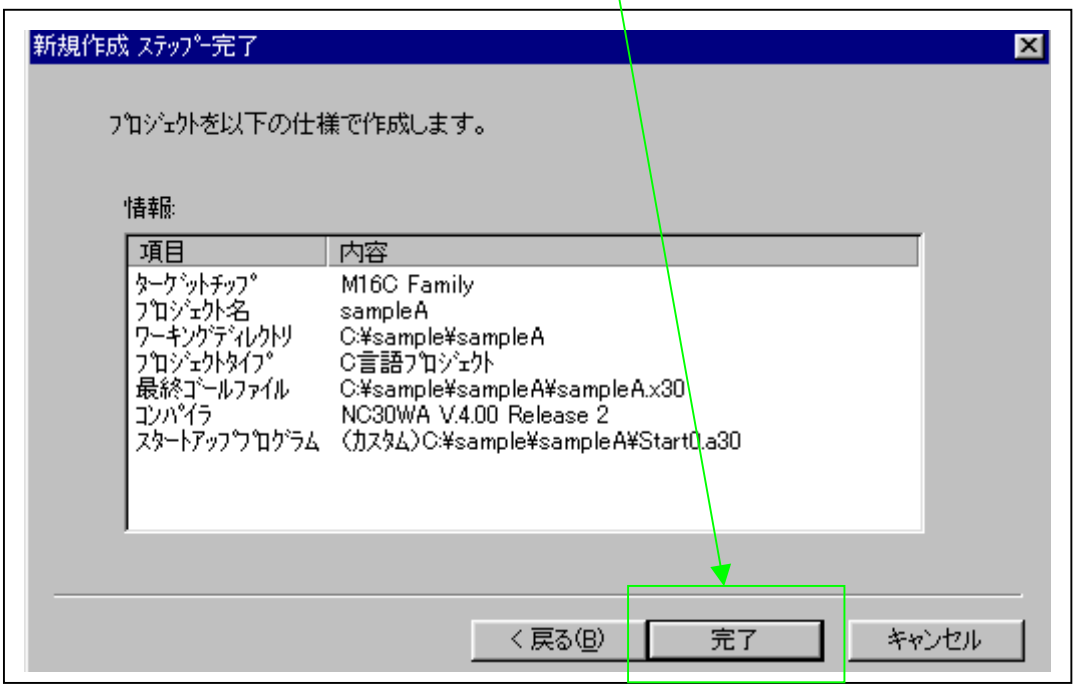
（オ）コンパイラ、スタートアッププログラムを指定します。スタートアッププログラムはどちらかを選択しなければなりません。今回は、start0.a30 を用意しましたが、用意していないときは「標準のスタートアッププログラムを使用する」を指定して下さい。この場合コンパイラに標準添付されているスタートアッププログラムがワーキングディレクトリにコピーされます。これを、ご自身のプログラムに合うように修正してください。



・ このボタンをクリックしてファイルの場所を指定します。

・ ファイルが指定できたら「次へ」をクリックします。

（カ）プロジェクトの確認画面が表示されます。「完了」をクリックして基本の設定は終了です。



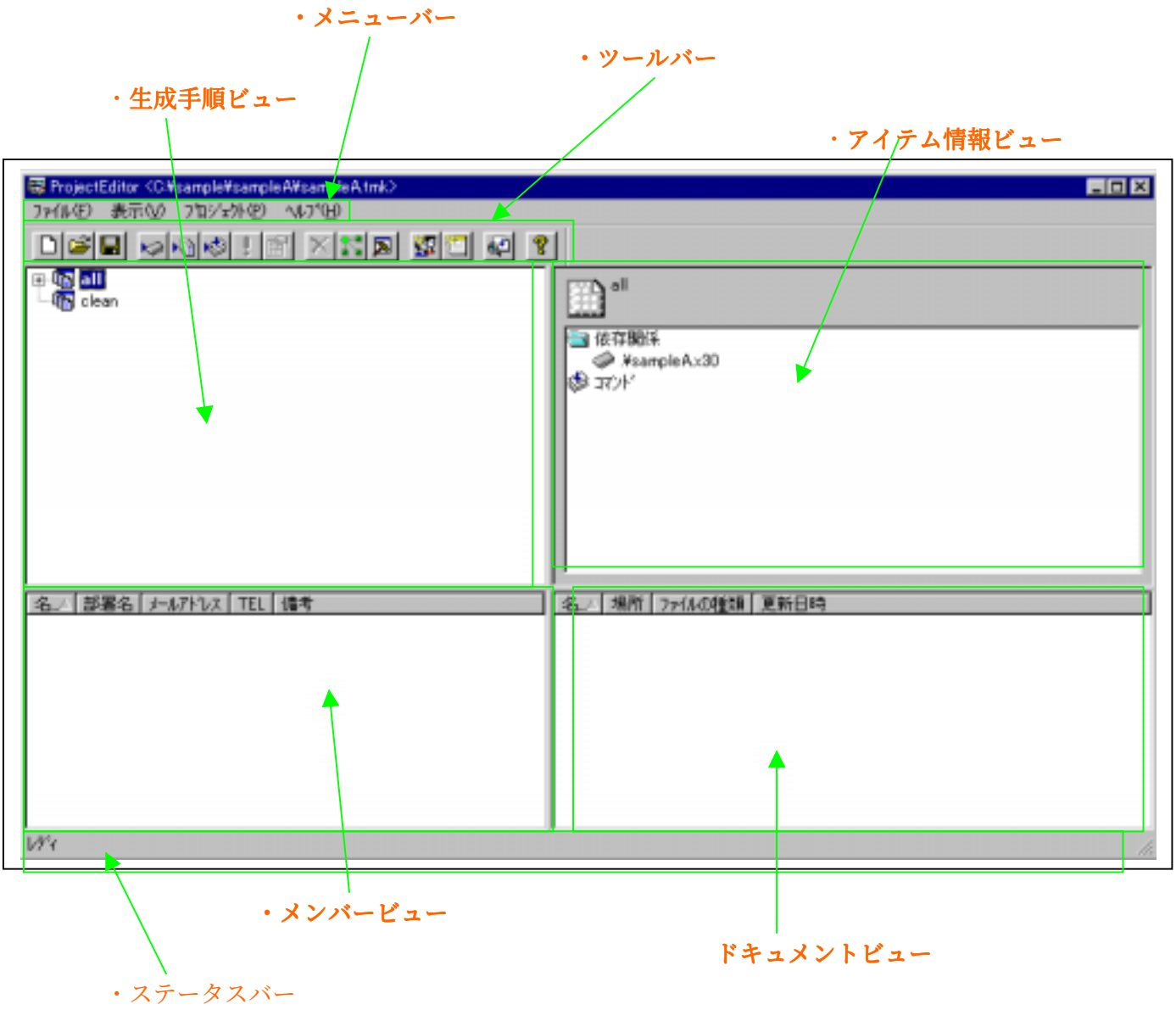
5.3.プロジェクトエディタ

次にプロジェクトエディタについて説明していきます。

プロジェクトエディタは、プロジェクトの表示編集を行うウインドウです。「5.2 基本プロジェクトの作成」で作成されたプロジェクトに含まれるソースファイルやコンパイル時のオプション、生成手順等の定義・変更を行います。

TM では make の記述方法を知らなくても、プロジェクトエディタでファイルを追加したり、コマンドオプションを指定するだけで、内部で自動的に makefile を生成します。そしてビルドボタンをクリックするだけで、make.exe を実行することになり、目的のファイルが生成されます。

5.3.1.ウインドウ構成



5.3.2.ウインドウ概要

（ア） メニュー

①ファイルメニュー

メニュー	メニュー項目	機能
ファイル(F)	プロジェクト新規作成(N)	新規にプロジェクトを作成する
	プロジェクトを開く(O)	既存のプロジェクトをオープンする
	上書き保存(S)	作業中のプロジェクトをファイルに保存（上書き）する
	名前を付けて保存(A)	作業中のプロジェクトをファイルに保存（名前を付けて）する
	makefile の出力(E)	作業中のプロジェクトを makefile に出力する
	最近使ったファイル	一度使用したプロジェクトの履歴から,プロジェクトをオープンする
	アプリケーションの終了(X)	プロジェクトエディタを終了する

②表示メニュー

メニュー	メニュー項目	機能
表示(V)	ツールバー(T)	ツールバーの表示・非表示を切り替える
	ステータスバー(S)	ステータスバーの表示・非表示を切り替える

③プロジェクトメニュー

メニュー	メニュー項目	機能
プロジェクト(P)	アイテムの編集(A)	生成手順で選択されたアイテムを編集する
	アイテムの追加（I）	生成手順にアイテムを追加する
	ファイルの追加（F）	生成手順にファイルを追加する
	コマンド（M）	生成手順のコマンドを編集する
	開く(O)	アイテムのプロパティを開く
	プロパティ(P)	アイテムのプロパティを表示する
	部分ビルド(C)	アイテムをビルドする
	削除(D)	アイテムを削除する
	マクロブラウザ(M)	マクロを表示（編集）する
	オプションブラウザ(P)	オプション表示（編集）する
	メンバ情報の追加(B)	プロジェクトにメンバ情報を追加する
	ドキュメントの追加(D)	プロジェクトにドキュメントを追加する
	情報(I)	作業中のプロジェクトの情報を表示する
	依存関係の更新(E)	生成手順の依存関係を更新する

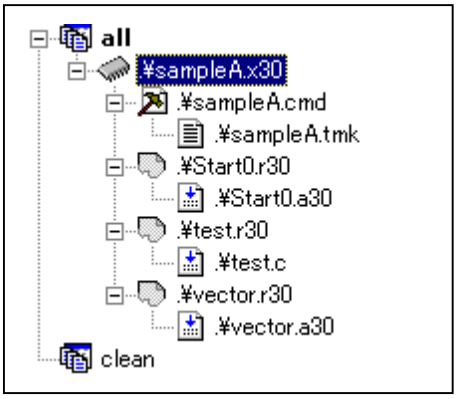
④ヘルプメニュー

メニュー	メニュー項目	機能
ヘルプ(H)	ヘルプ(H)	ＴＭのヘルプ及びプロジェクトで使用しているツールのヘルプを表示します
	電子マニュアル(M)	ＴＭの電子マニュアル及びプロジェクトで使用しているツールの電子マニュアルを表示する
	ProjectEditor のバージョン情報(A)	プロジェクトエディタのバージョン情報を表示する

⑤ツールバー

ボタン	ボタン名	内容
	新規プロジェクト	新規にプロジェクトを作成する
	プロジェクトを開く	既存のプロジェクトをオープンする
	プロジェクトの上書き保存	作業中のプロジェクトをファイルに保存（上書き）する
	アイテムの追加	生成手順にアイテムを追加する
	ファイルの追加	生成手順にファイルを追加する
	コマンド編集	生成手順のコマンドを編集する
	開く	アイテムを関連付けされたアプリケーションで開く
	プロパティ	アイテムのプロパティを表示する
	削除	アイテムを削除する
	マクロブラウザ	マクロを表示（編集）する
	オプションブラウザ	オプションを表示（編集）する
	メンバーの情報の追加	プロジェクトにメンバー情報を追加する
	ドキュメントの追加	プロジェクトにドキュメントを追加する
	依存関係の更新	生成手順の依存関係を更新する
	バージョン情報	プログラム情報、バージョン情報そして著作権を表示

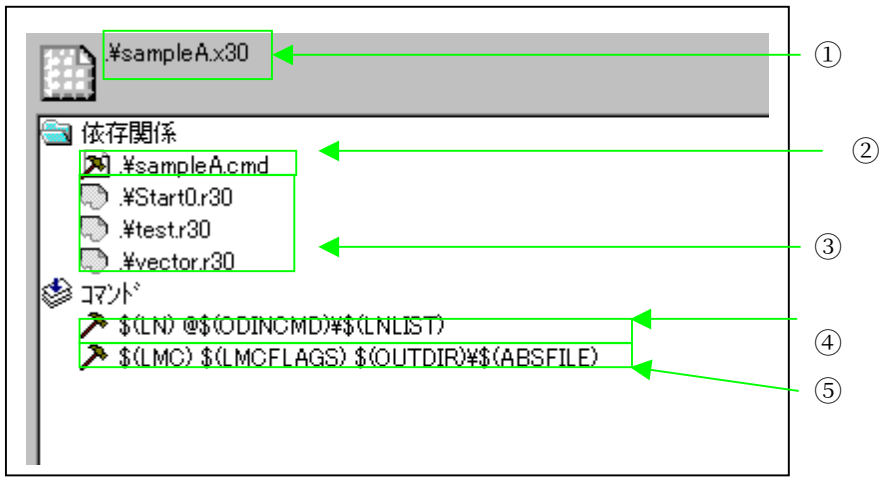
（イ）生成手順ビュー



生成手順ビューは、プロジェクトの生成手順（依存関係）情報をツリー状に示します。
「all」は、プロジェクトをビルドする（オブジェクトプログラムを作成する）ためのアイテムです。
「clean」は、生成されたオブジェクトを削除するためのアイテムです。

（ウ）アイテム情報ビュー

アイテムビューは、生成手順ビューで選択されたアイテム情報の詳細を表示します。以下にアイテムビューの概略図を示します。



① 選択されているファイルです。

② リンクする為のコマンドが記述されているファイルです。

③ ①を生成する為のソースとなるファイルです。

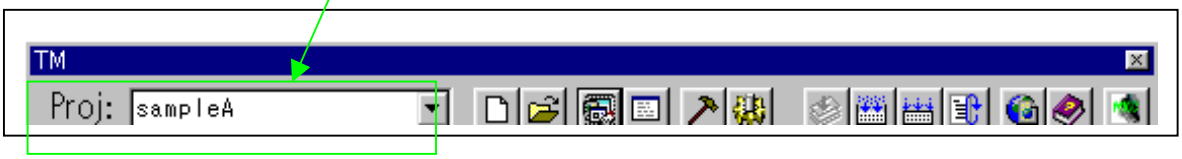
④ ①を生成する為のコマンドです。

⑤ flashstart（フラッシュＲＯＭに書き込むためのソフト）で使用するモトローラ S フォーマットのファイルを作成するためのコマンドです。「5.6. モトローラＳフォーマットのファイル作成」で詳しく説明します。

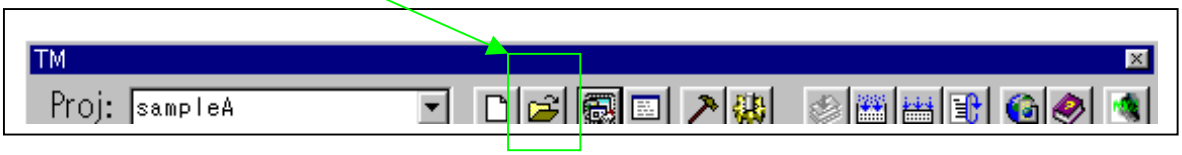
5.4.ファイルの追加

では、プロジェクトエディタでファイルを追加する方法を説明していきます。
まず、「5.2 基本プロジェクトの作成」で作成した、内容を確認します。

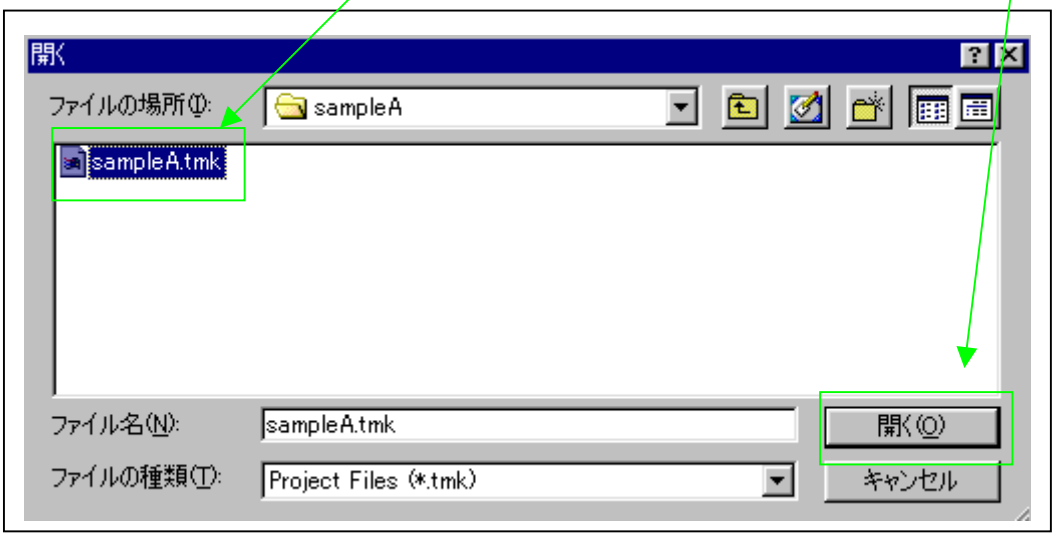
（ア）TMを立ち上げます。プロジェクト名には、前回開いていたプロジェクト名が入っています。このプロジェクトを開くときにはそのままプロジェクトエディタ起動ボタンをクリックします。別のプロジェクトを開く場合は、プロジェクトオープンボタンをクリックします。



（新しいプロジェクトを開く場合）
①プロジェクトオープンボタンをクリックします。

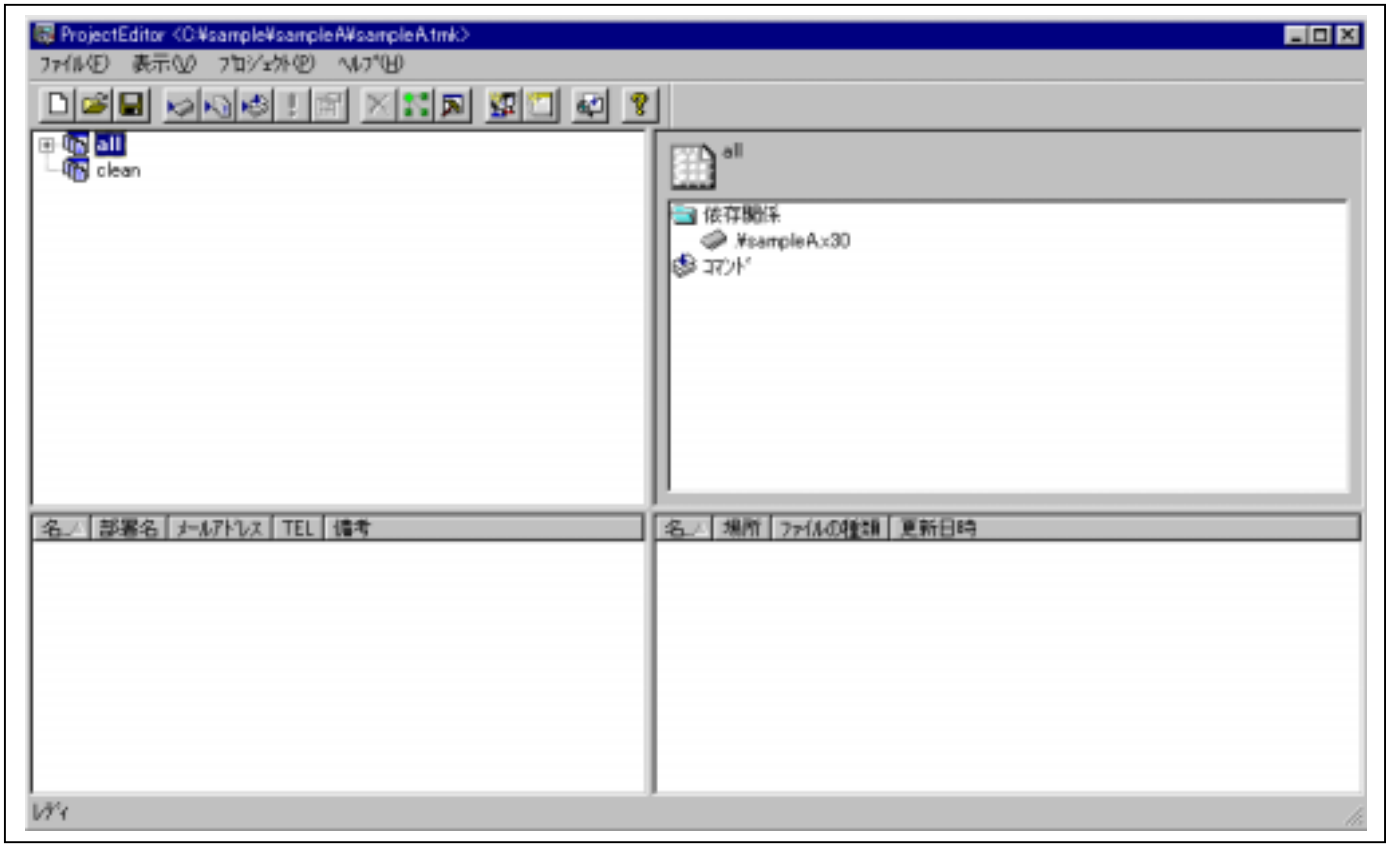


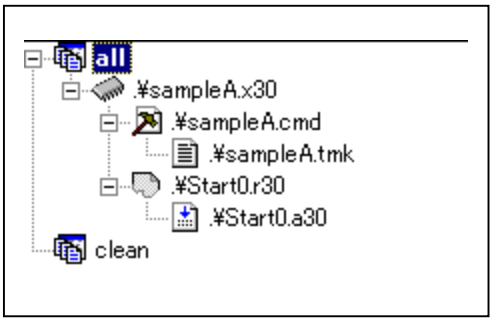
②ファイルの場所を指定し、「sampleA.tmk」というプロジェクトファイルを選択し、開きます。



③プロジェクトエディタを開きます。

（イ）プロジェクトファイルが開きました。



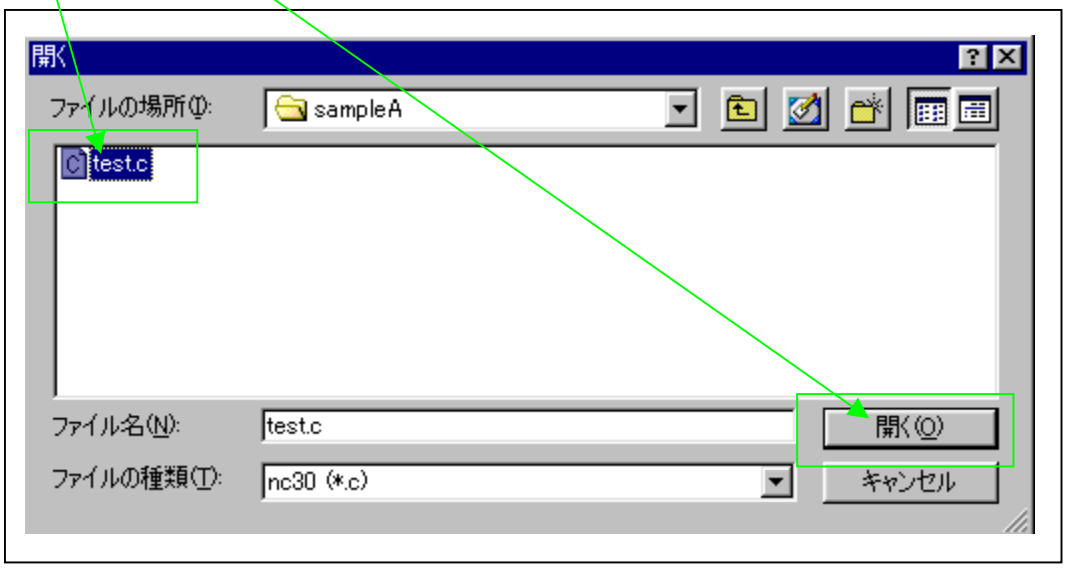


⊞のマークをクリックして全てのファイルを表示してみましょう。
ここにファイルの依存関係を考慮してファイルを追加していきます。プログラムを構成するためにまだ登録していない 2 つのファイル test.c と vector.a30 を追加します。

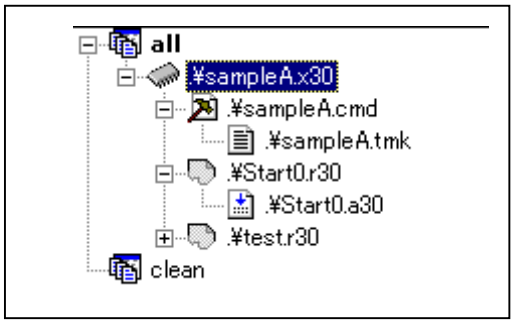
（ウ）ファイルの追加

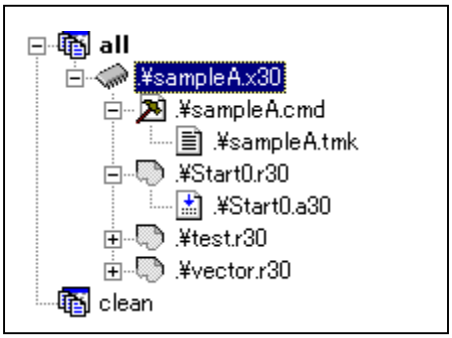
最終ファイルである「sampleA.x30」を選択し右クリックから「アイテム編集」→「ファイルの追加」を選択します。

Test.c を選択し、「開く」をクリックします。

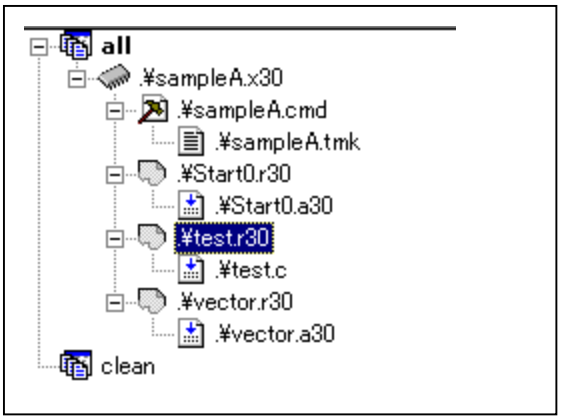


中間ファイルとして test.r30 が設定されました。同様に、vector.a30 も追加します。





 をクリックして全てのファイルを表示すると次のようになります。

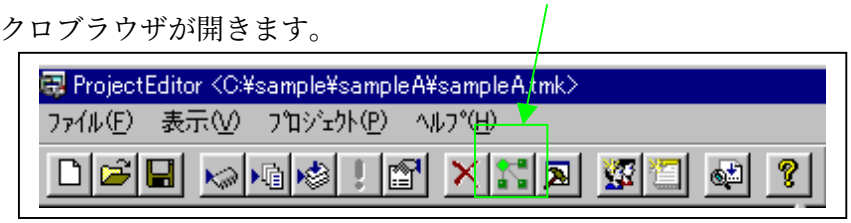


これで、必要なファイルが追加できました。

5.5.コマンド

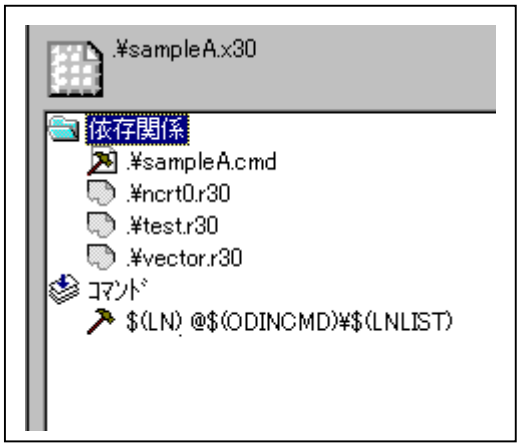
プロジェクトエディタの、アイテム情報ビューに表示されるコマンドは、マクロで書かれています。このマクロには、あらかじめ設定されているもの、プロジェクトの新規作成ウイザードで作成されるものがあり、もちろんユーザーが後から追加、削除することもできます。

現在プロジェクトエディタで開かれているプロジェクトに設定されているマクロは、マクロブラウザで確認することができます。マクロブラウザボタンをクリックしてください。
マクロブラウザが開きます。



新しくマクロを追加したい場合は、新規ボタンをクリックします。既に存在するマクロを編集・削除する場合にはマクロを指定すれば、ボタンがアクティブになり操作できるようになります。

このマクロに照らし合わせて、アイテム情報ビューのコマンドをもう一度見てみましょう。



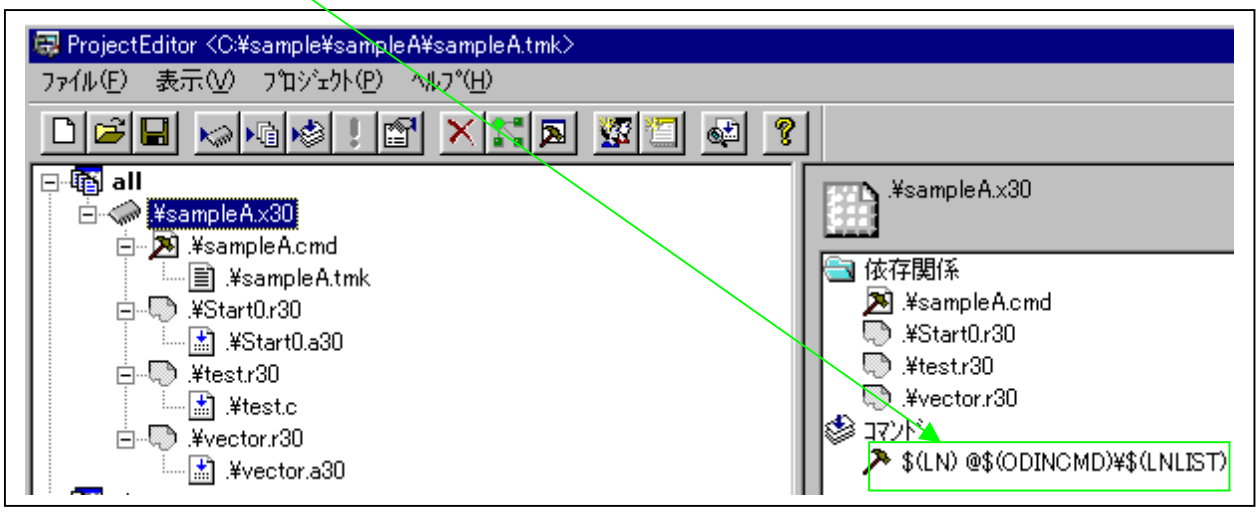
このビューは sampleA.x30 のものです。
コマンドの中の表示はそれぞれマクロ定義をもとに記述されています。ここで、全てのマクロを理解する為には、もう少し make の記述を調べていかなければなりません。
ＴＭでは、ファイルを追加すれば自動的にコマンドを記述します。ですから、ここでは各コマンドラインがどのソフトを起動するためのものなのかということだけ知っておいて頂ければいいと思います。

$(LN)で始まる行は、ln30 の起動コマンドです。
$(CC)で始まる行は、nc30 の起動コマンドです。
$(AS)で始まる行は、as30 の起動コマンドです。
$(LMC)で始まる行は、lmc30 の起動コマンドです。

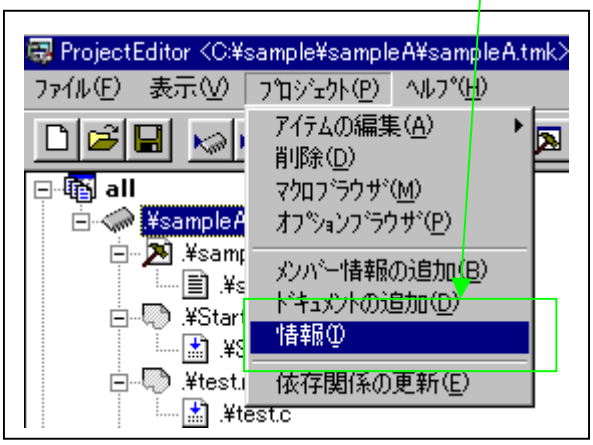
5.6.モトローラＳフォーマットのファイル作成

OAKS16 の開発環境では、デバッグ終了後フラッシュＲＯＭにプログラムを書き込むために、flashstart というソフトを使用します。この flashstart では、モトローラＳフォーマットのファイルを扱う為、”.x30”という拡張子のアブソリュートモジュールファイルを”.mot”という拡張子のファイルに変換する必要があります。この作業を行うのが「lmc30」です。
TM で lmc30 を使用するためには、「ツールの使用」の設定を行います。この後に具体的な手順を示していきます。

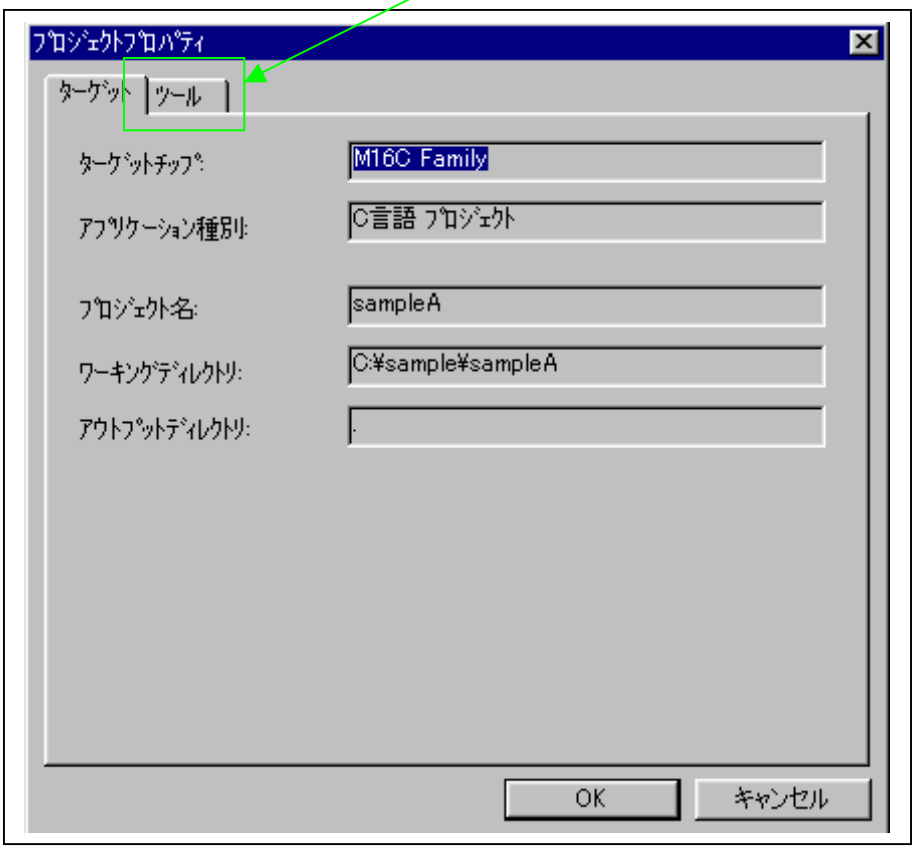
（ア） プロジェクトエディタを開きます。ここで、sampleA.x30 のコマンドを確認すると、ln30 の起動コマンドだけが表示されています。



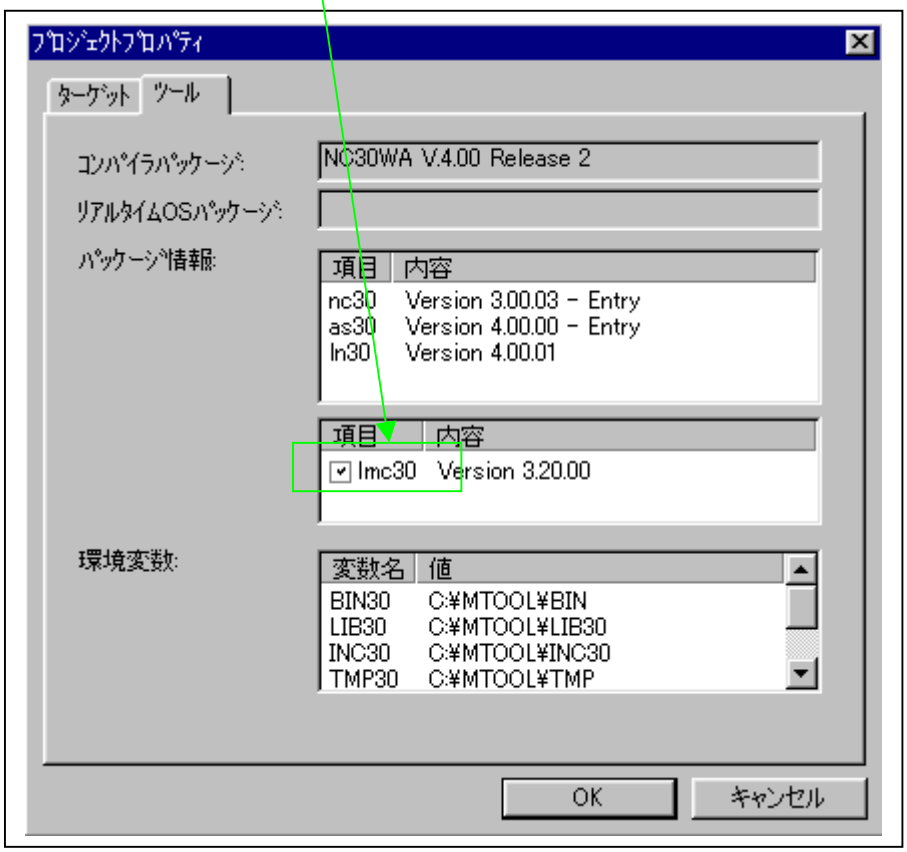
（イ）ここでプロジェクトメニューから「情報」を選択します。



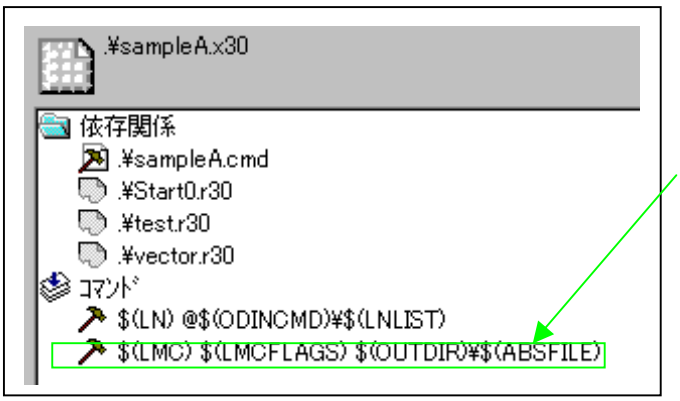
（イ） プロジェクトプロパティが開きます。「ツール」をクリックします。



（エ）lmc30 にチェックし、「ＯＫ」をクリックします。



これで、アブソリュートモジュールファイルと一緒に、モトローラＳフォーマットのファイルが作成されることになります。lmc30 を追加しても、生成手順ビューの表示は変わりません。アイテム情報ビューのコマンドに lmc30 の起動コマンドが追加されるだけです。このままでは表示は変更されませんので生成手順ビューで sampleA.x30 以外のファイルを一度クリックしてから再び sampleA.x30 をクリックして下さい。

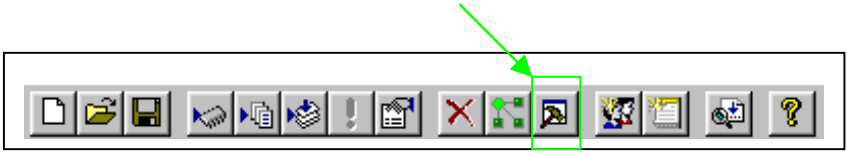


lmc30 のコマンド追加が追加されました。

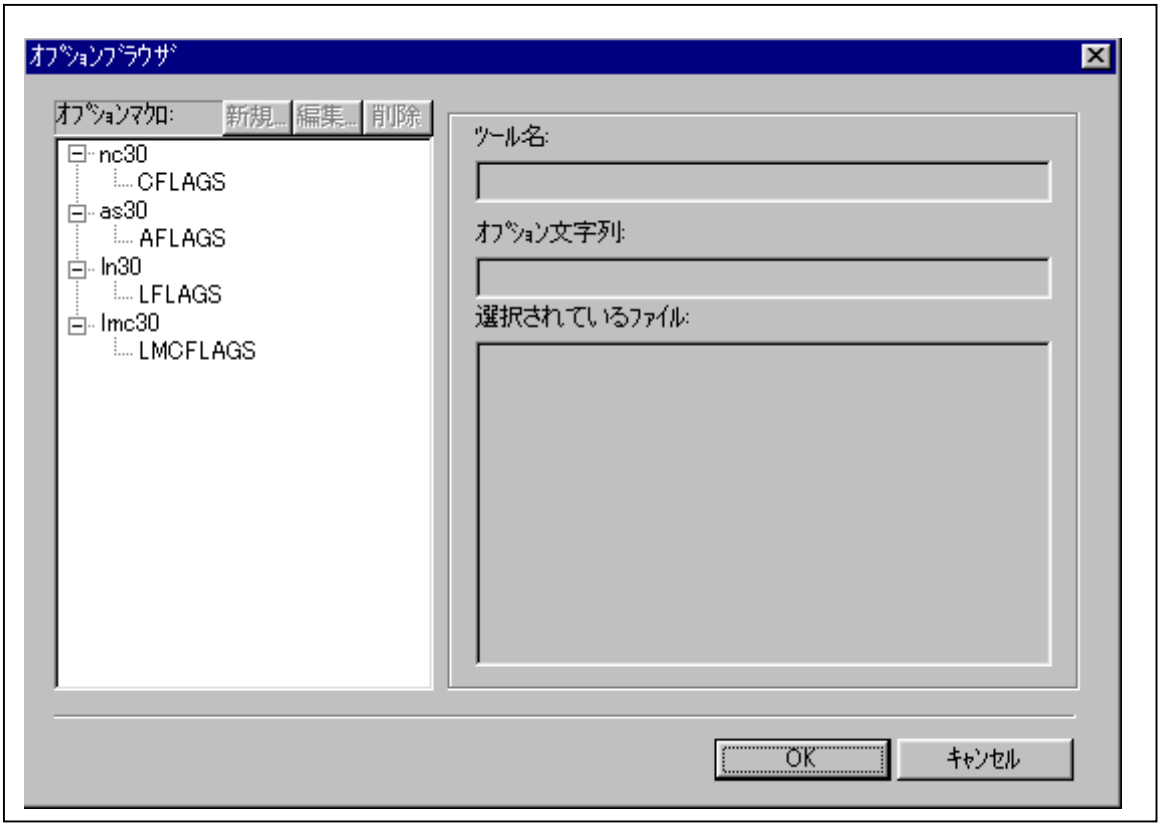
5.7.オプションの指定

プロジェクトの作成と、ファイルの登録により、プログラムの開発の手順は決定されました。ここではその過程で実行されるそれぞれのコマンドに対するオプションの指定方法を説明します。

（ア） ツールバーの「オプションブラウザ」ボタンをクリックします。



（イ） オプションブラウザの設定ダイアログが表示されます。

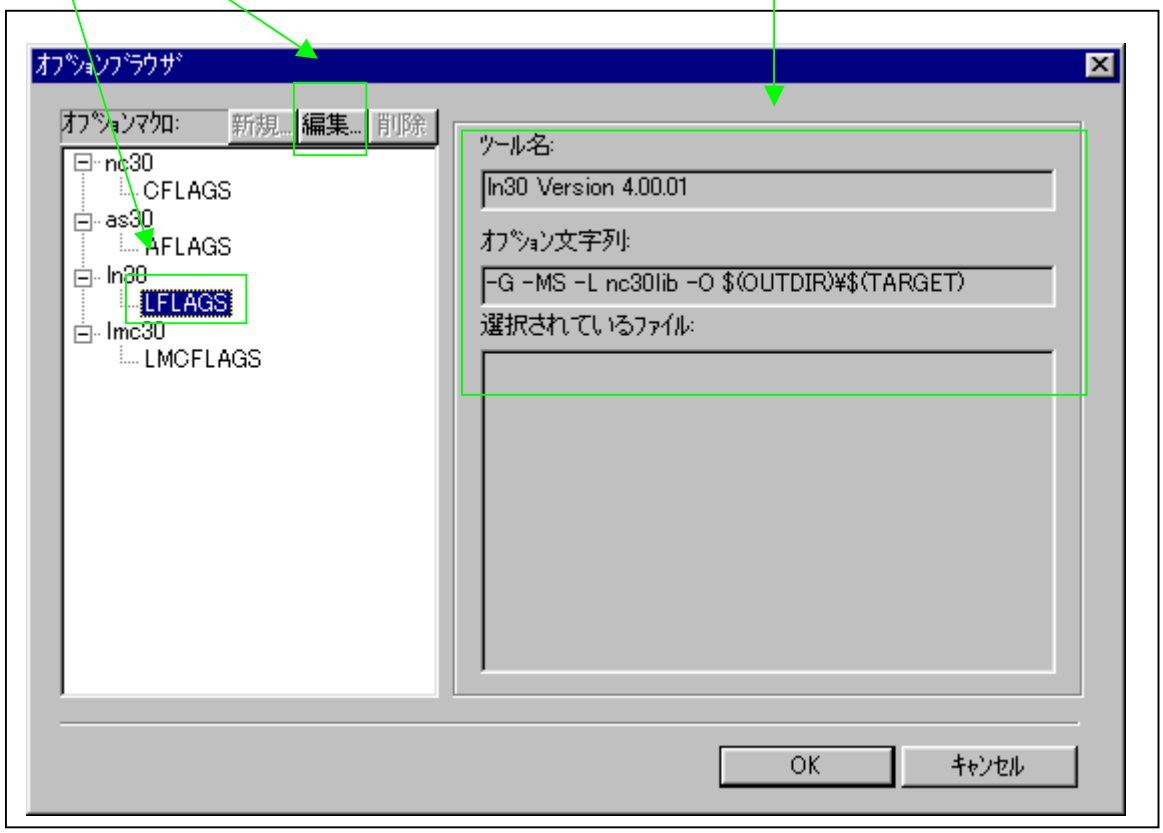


ここで、コンパイラ(nc30)、アセンブラ(as30)、リンカ(ln30)、ロードモジュールコンバータ(lmc30)の起動オプションを指定していきます。

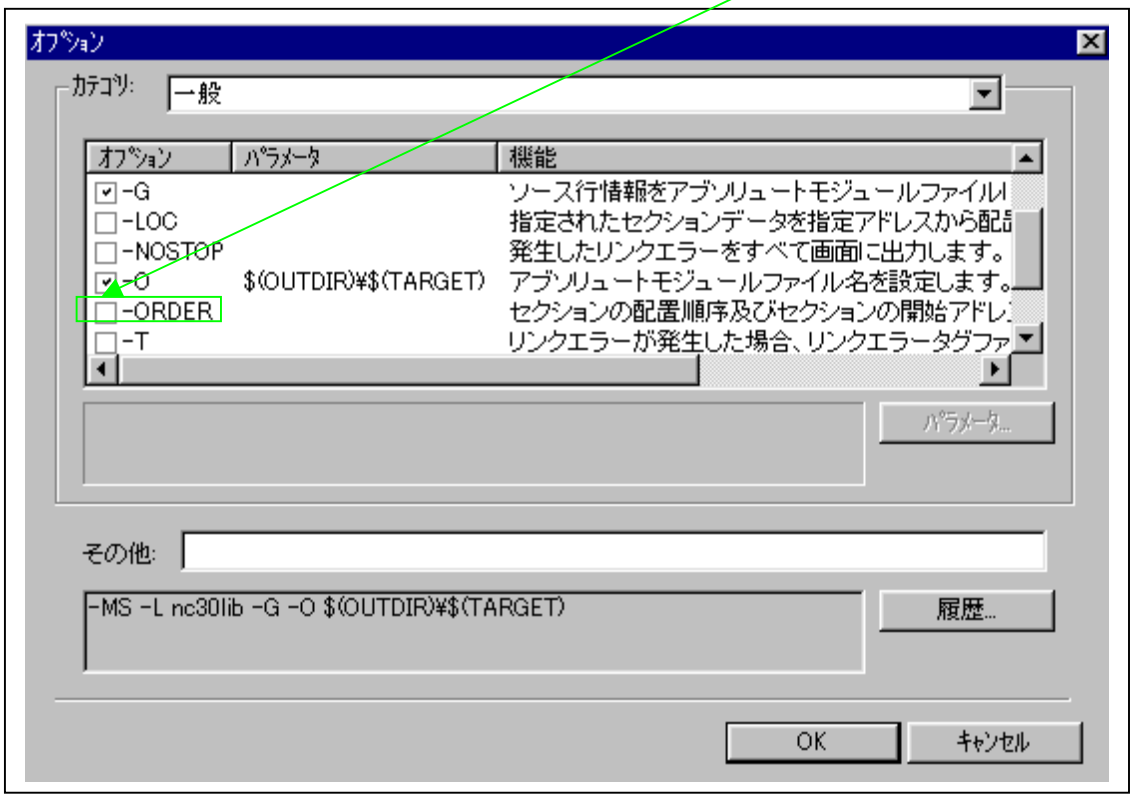
それぞれのオプションはあらかじめ設定してあるものもありますが、追加削除が可能です。

また、ファイルごとにオプションを変えることも可能です。

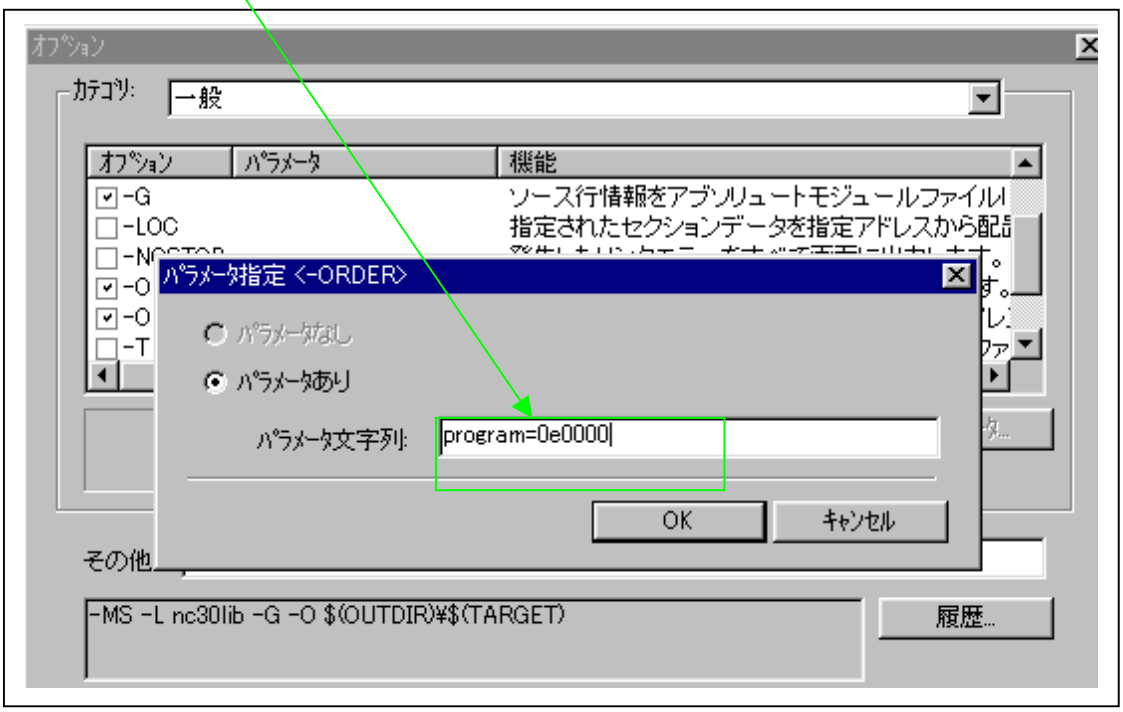
フラグを選択するとツール名と既に指定してあるオプション文字列が表示されます。ここで編集ボタンをクリックするとオプションダイアログが開きます。

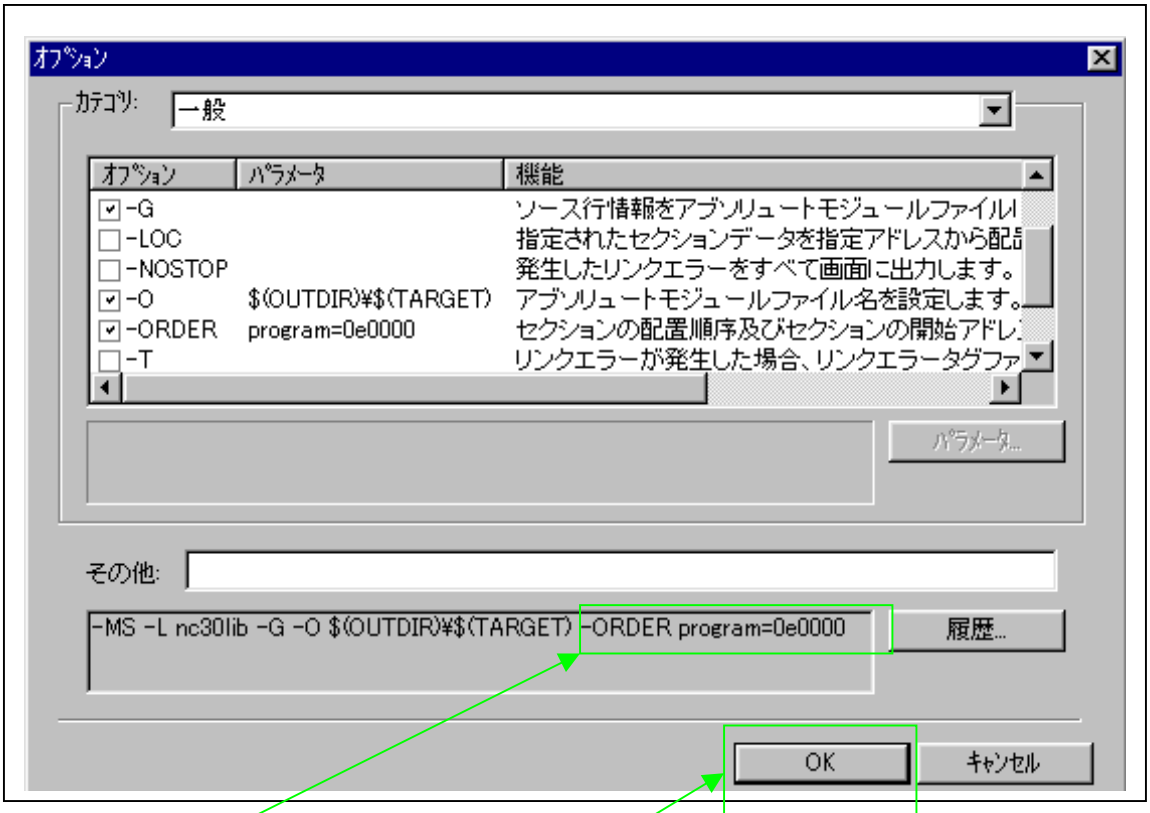


オプションのカテゴリは、ツール（nc30、as30 など）ごとに異なった分類で表示されます。カテゴリを選択し、オプションの先頭にある□にチェックするだけで設定は OK です。
ここでは、リンクするときのプログラムの先頭アドレスを指定する為、ORDER オプションをチェックしてみます。



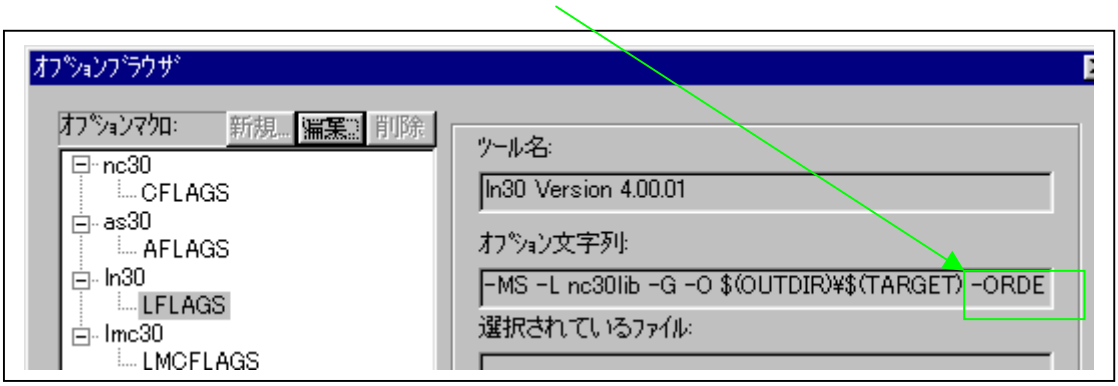
ORDER はパラメータを必要とするオプションなのでパラメータ指定の画面が出ます。プログラムセクションを 0e0000ｈ番地（ユーザーROM の先頭番地）に指定し「OK」をクリックします。





履歴に設定したオプションが追加されます。「ＯＫ」をクリックしてください。

LFLAGS のオプション文字列に-ORDER が追加されました。



注意：
プロジェクトエディタでプロジェクトを作成する場合、C 言語ソースファイルを追加したときのコンパイラ nc30 のオプションは“ -c ”になります。（リロケータブルオブジェクトファイル（拡張子 r30）を出力する。）これを“ -S ”オプションに変更して a30 ファイルを出力し、その後 as30 でアセンブルするということはできません。

5.8.依存関係の更新

このボタンでプロジェクトに登録されたソースファイルの依存関係を更新します。これにより、ソースファイルに記述されたインクルードファイルが認識され、自動的にプロジェクトの生成手順に追加されます。プロジェクト作成終了時、ソースファイルの追加、編集終了後など必要なタイミングで依存関係の更新を行なってください。依存関係の更新を行なう為には、ツールバーの「依存関係の更新」ボタンをクリックしてください。

以上で、プロジェクトの作成に必要な作業が終了しました。この段階でプロジェクトエディタの終了やビルドを行なうとプロジェクトの保存確認のメッセージが表示されます。保存するように指定しますと、プロジェクトエディタで編集してきたことがプロジェクトファイルに保存されます。

6.ビルド

6.1.ビルドの種類

ビルドは、プロジェクトエディタで選択されているプロジェクト（表示されている内容）のコマンドを実行する為のものです。ＴＭのビルドは３種類あります。

①ビルド：make の実行と同様に、オブジェクトファイルが存在しないとき、もしくはソースファイルの変更が行われた場合にコマンドを実行します。

②リビルド:一度､clean に記述されているコマンドを実行してから､ビルドを行います。（clean によって、中間生成ファイルと、オブジェクトファイルが削除されますので、必ずプログラム作成の為の全てのコマンドが実行されます。）

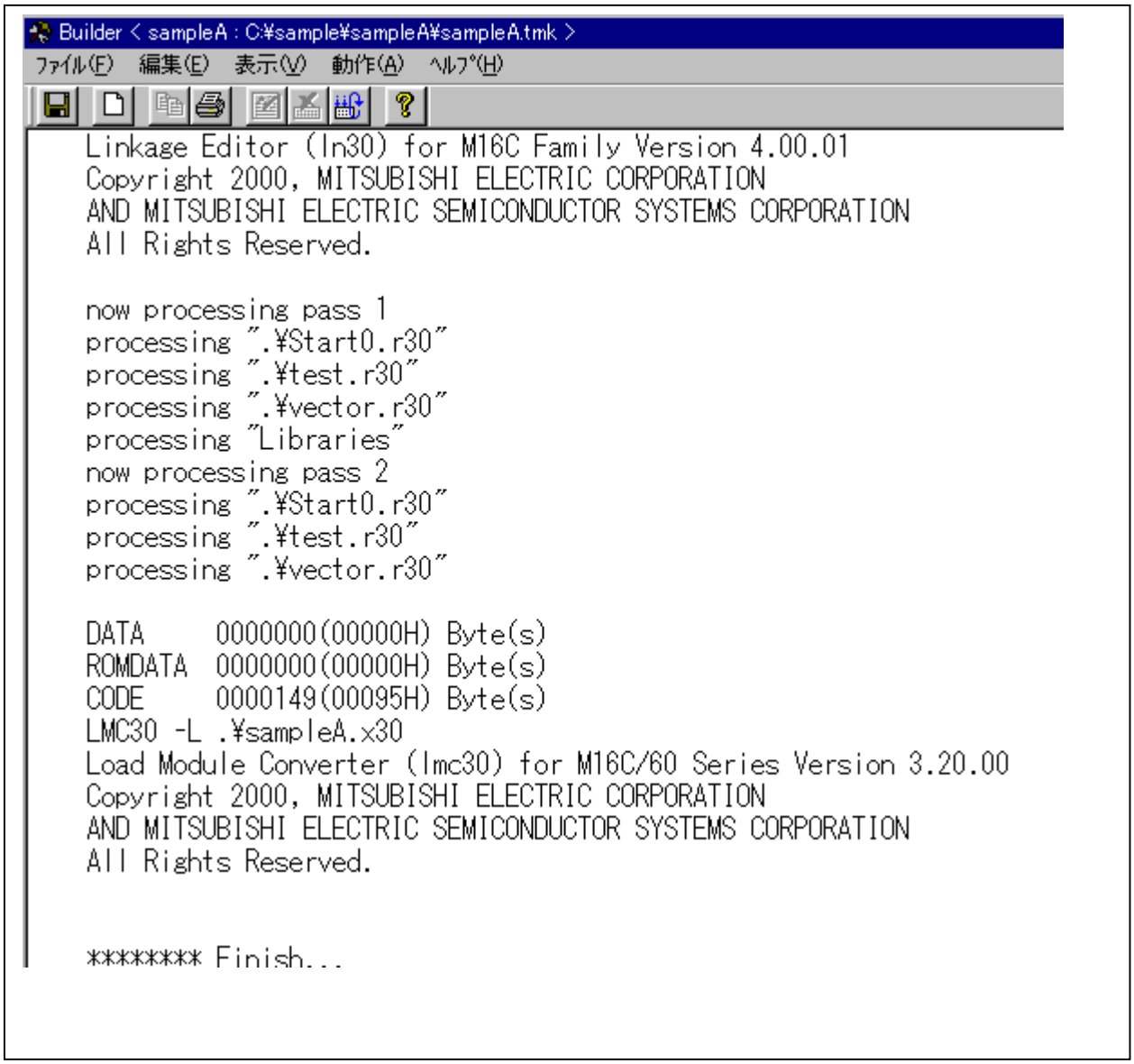
③部分ビルド：プロジェクトエディタで選択したアイテムをビルドします。

《ビルドボタン》

6.2.ビルドの実行例

プロジェクトファイル名が選択されているところで「ビルド」ボタンをクリックすると

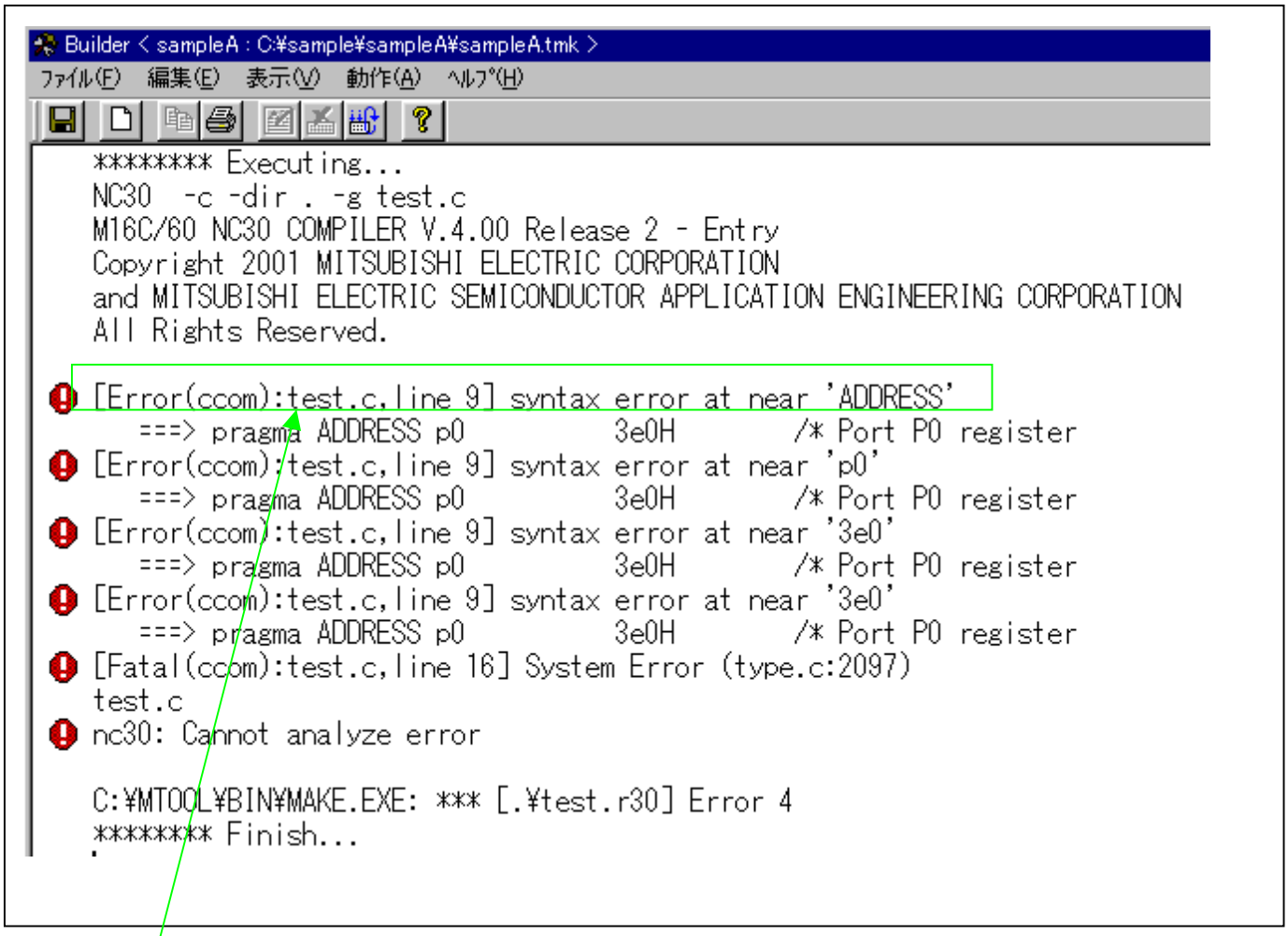
ビルダーウインドウが開きます。



6.3.ビルドでコンパイラの ERR が発生した場合

ビルドを行うと、プロジェクトエディタで設定された make の手順により、コンパイル、アセンブル、リンクが順番に行われていきます。その途中で、ソースファイルの記述ミスなどが発生した場合、ビルダーウインドウから、ソースファイルをエディタで開き、ＥＲＲ行にジャンプすることができます。

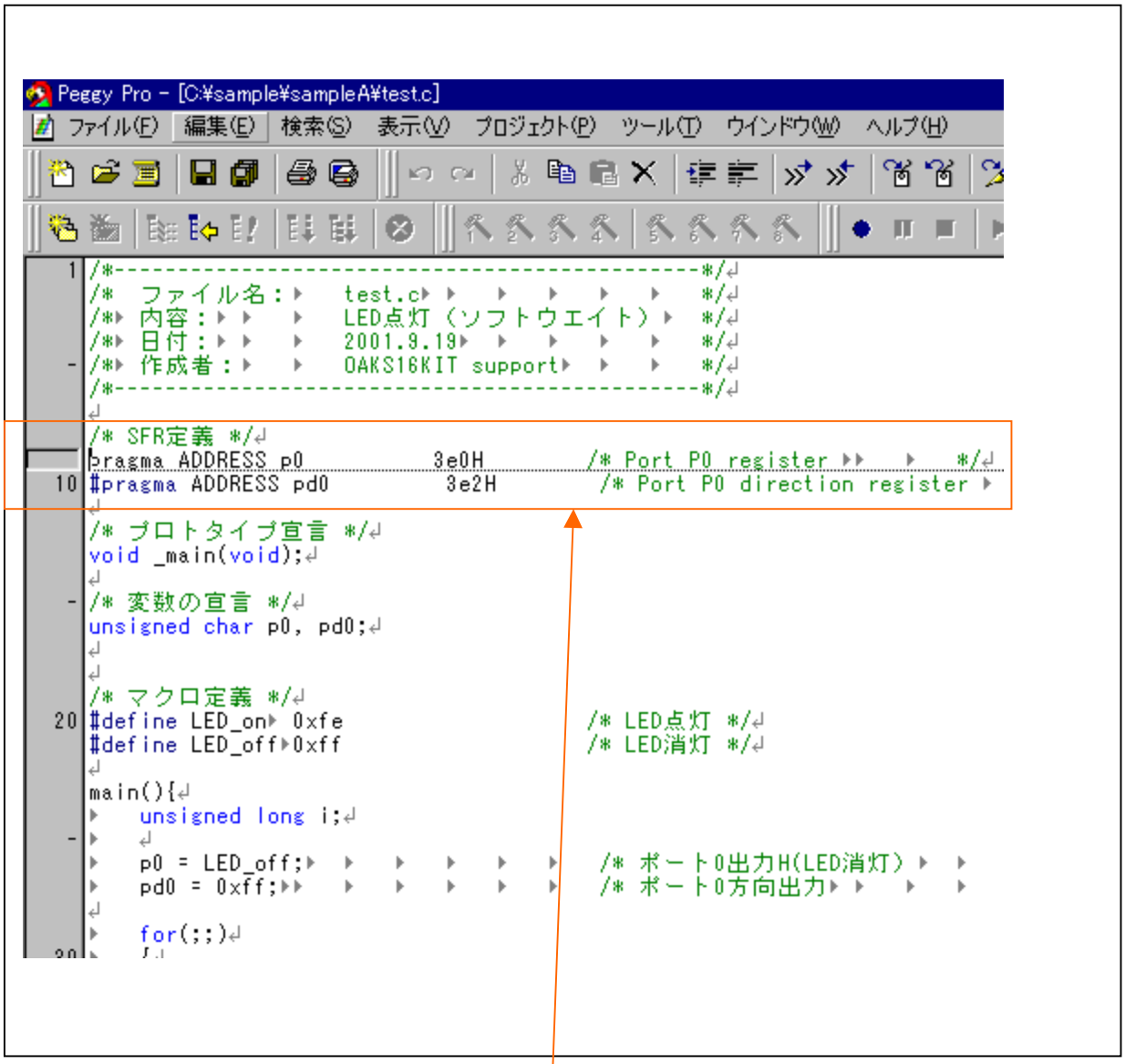
例として、test.c の中の 9 行目「#programa」の「#」をはずして、記述 ERR が出たときの対応を示します。



ビルダーが ERR 表示を出し、ビルドが終了しました。

ERR 行をダブルクリックすると、登録されたエディタで ERR の発生したファイルを開き、カーソルをその行に移動します。

9 行目の ERR をダブルクリックしてみましょう。



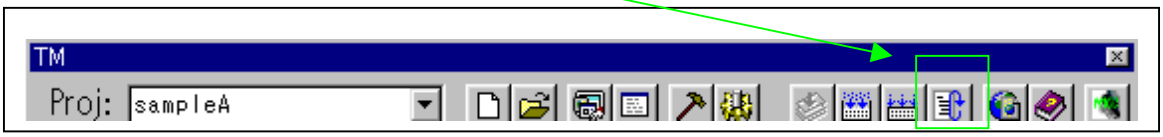
この例では、エディタ（peggypro）が、開いた例です。ERR 行(9 行目)にカーソルが移動しました。

ここで、ERR を修正し、再び TM で、ビルドを行ってください。

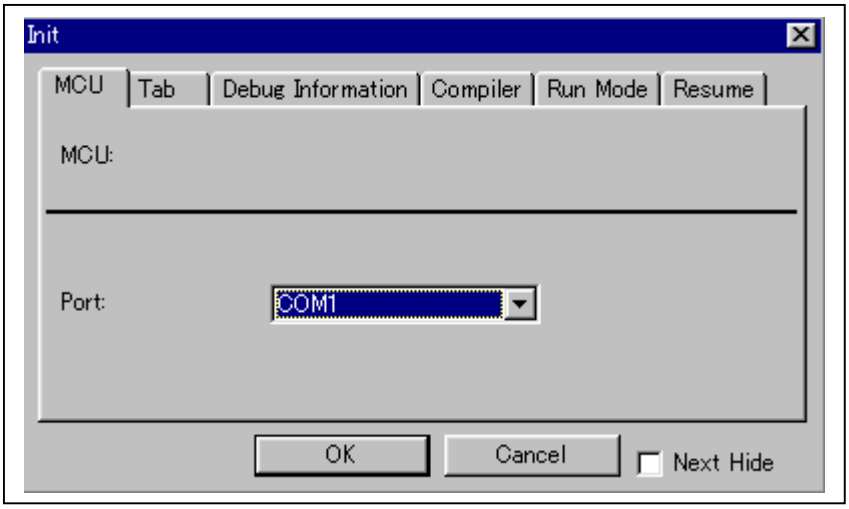
7.デバッガの起動

以上で、オブジェクトプログラムの作成までが終了です。
この後、TM のプロジェクトバーから、デバッガ（KD30）を起動し、作業を進めてください。

（ア） デバッガの起動
「デバッグボタン」をクリックします。

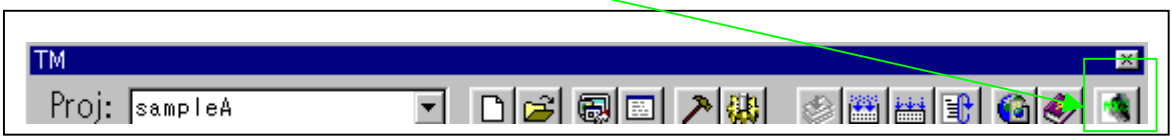


デバッガの初期画面が表示されます。この後の操作方法は、デバッガのマニュアルを参照して下さい。

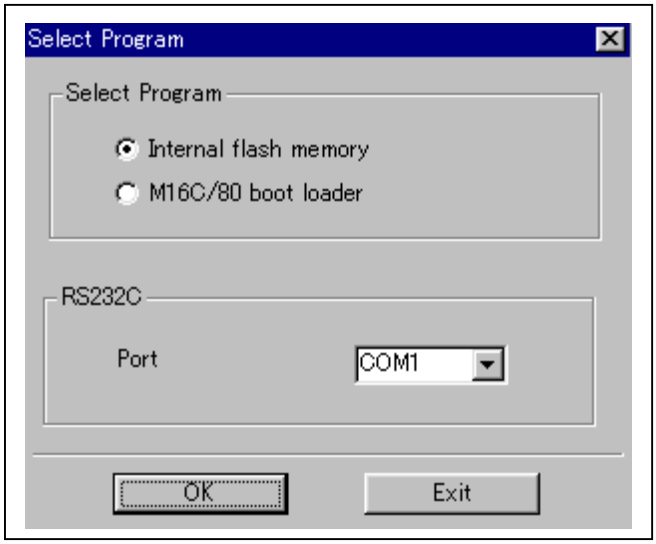


8.フラッシュＲＯＭライタの起動

「フラッシュライタ」ボタンをクリックします。」



デバッガと同様に初期画面が表示されます。



このあとは、フラッシュスタートのマニュアルを参考にして、操作を進めてください。

ＯＡＫＳ16 キット
ＯＡＫＳ16 で TM をお使いになる方のために（Ver. 1.05）

資料番号：ＯＡＫＳ16 でＴＭをお使いになる方のために（Ver. 1.05）
発 行 所：オークス電子株式会社
〒101-0025　東京都千代田区佐久間町 3 丁目２１番地（第一千代田ビル３F）
ＴＥＬ：０３－３８６３－１１２１　　ＦＡＸ：０３－３８６３－１１３０
ホームページ　　http://www.oaks-ele.com