



MPLAB v3.40 プロジェクト: チュートリアル

はじめに

MPLAB v3.40 のプロジェクト・マネージャーは、機能拡張により、複数のファイルをサポートできるようになりました。MPLAB v3.31 やそれ以前のバージョンで作成したプロジェクトは、オープン時に、自動的に MPLAB v3.40 フォーマットのプロジェクトに変換されます。変換後のプロジェクトは、MPLAB の前バージョンでは再びオープンすることができません。

ハイライト

このチュートリアルでは、MPLAB プロジェクトの以下の機能を学習します。

MPLAB プロジェクトの概要

- 1 つの MPASM ソースファイルを使用したプロジェクトの作成
- プロジェクト作成なしの単独 MPASM ソースファイルのコンパイル
- MPLINK を使用した複数の MPASM ソースファイルのプロジェクトの作成
- MPLAB-C17 を使用したプロジェクトの作成
- Hi-Tech PIC C を使用したプロジェクトの作成

このタスクを実行するために、MPLAB の以下の機能を使用します。

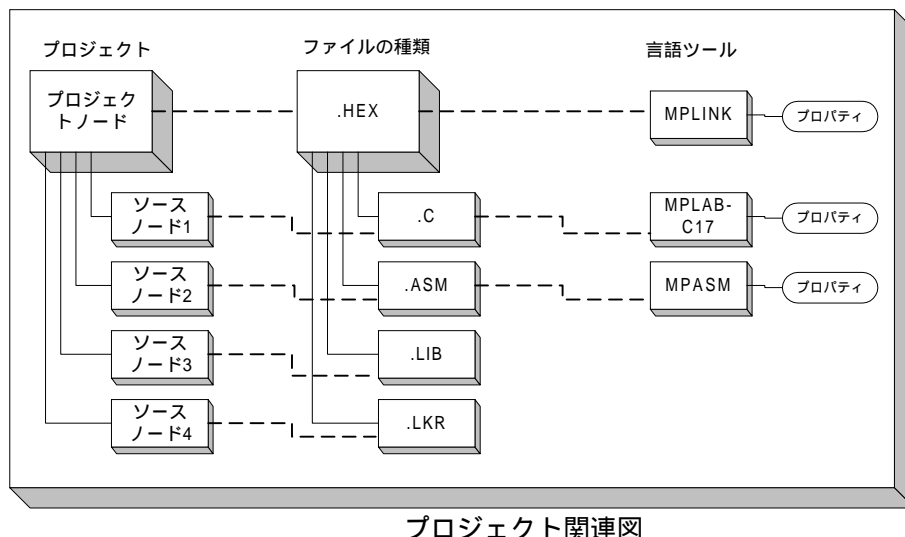
- 言語ツールのインストール
- 新規プロジェクト
- プロジェクトへのノードの追加
- プロジェクトノードプロパティの設定
- プロジェクトの作成・構築
- プロジェクトウィンドウ

用語集

プロジェクト	MPLAB のプロジェクトは、1 つのアプリケーションを作成するためのファイル群です。構築ツールとのリンクも含まれます。
構築ツール	通常はコンパイラ、アセンブラ、または、リンカ。
ノード	いくつかの種類ファイル。プロジェクトのコンポーネント。
プロジェクトノード	プロジェクトの最上位レベル。その他すべてのファイルは、このノードを構築するために使用します。
ソースノード	アプリケーションを構築するためのコンポーネントファイルの 1 つ。これは、アセンブラのソースファイル、C 言語のソースファイル、あらかじめコンパイル済のオブジェクトファイル、ライブラリ、またはリンカのスクリプトのどれかです。。
プロジェクトの作成	ソースコードまたはプロジェクトオプションに変更があった場合のみ、すべてのソースノードをコンパイル、アセンブル、リンクしてアプリケーションを構築すること。
プロジェクトの構築	無条件で、すべてのソースノードをコンパイル、アセンブル、およびリンクをすること。

MPLAB プロジェクトの概要

MPLAB のプロジェクトはアプリケーションの構築に必要なファイル群です。各種構築ツールとファイル群の関連付けも行います。プロジェクトは、プロジェクトノードと、1つまたは複数のソースノードで構成されます。ソースノードの代表的なものは、MPASM ソースファイル、MPLAB-C17 ソースファイル、コンパイル済のライブラリと、オブジェクトファイル、リンカスクリプトです。通常、プロジェクトは、メインソースファイルと同じディレクトリに格納します。



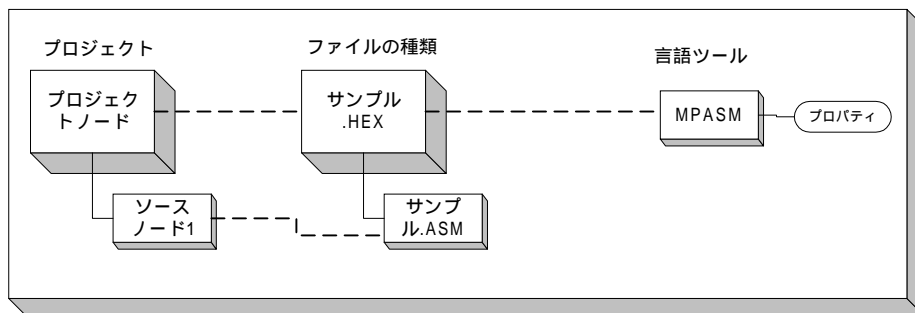
プロジェクトの各ノードは、それぞれ言語ツールで関連付けされます。プロジェクト作成時に、必要ない場合を除いて、言語ツールを割り当てなければいけません。例えば、MPASM を1つのソースファイルで使用している場合（次に出てくる最初のチュートリアルを参照）、プロジェクトノードがすでに MPASM にセットアップされているので、ソースファイルノードに言語ツールを割り当てる必要がありません。同様に、MPLAB-C17 用または MPASM 用のコンパイル済ライブラリ、オブジェクトファイル、リンカのスクリプトは、常に MPLINK により作成されるためツールを割り当てる必要がありません。

それぞれの言語ツールには複数のオプションがあります。オプションは DOS 上で起動した場合はコマンドラインスイッチで設定します。オプションは各編集可能なノードのノードプロパティとして表示されます。オプションのスイッチは .MAP ファイル生成をする/しない、適切な .HEX フォーマットの選択、警告メッセージの表示/非表示などを制御します。オプションは、1つのプロジェクトの各種ソースノードに対して別々に設定可能ですが、ほとんどの場合は同じ設定となります。ノードプロパティダイアログは各種ツールのコマンドラインスイッチに対応しており、初めて開く時、ダイアログはデフォルト値を表示します。これらのコマンドラインスイッチについては MPASM と MPLAB-C17 のユーザーズガイドを参照してください。

MPLAB のプロジェクトマネージャの構築は MAKE で行います。このツールは、そのプロジェクトの全アプリケーションを作成するソースファイルの更新日時を確認して、プロジェクトが再リンクされる時に、どのコンポーネントが変更され、再コンパイルまたは再アセンブルが必要であるかを判定します。

1つのMPASM ソースファイルによるプロジェクトの作成

1つのソースファイルのみのプロジェクト、または以前の方法(MPLAB バージョン 3.31 以前)を使用するプロジェクトを作成するには、次のステップに従います。ただし、ソースファイルは、他のファイルを #include できます。



1つのMPASM ソースファイル・チュートリアルプロジェクト関連図

開発モードの設定

アプリケーションに対して適切な開発モードを選択します。ここでは、Options>Development Mode の設定を MPLAB-SIM シミュレータにして、16F84 PICmicro を選択します。

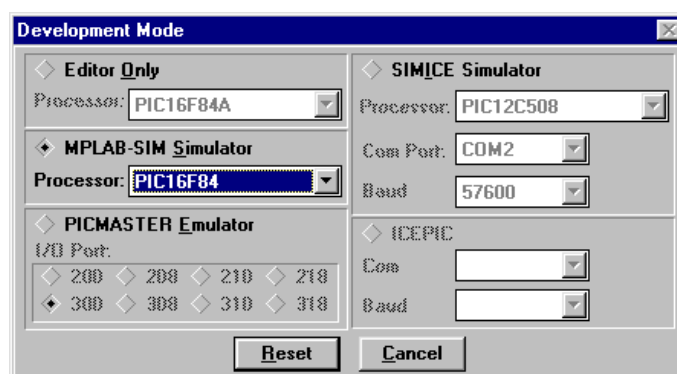


図 1

新規のプロジェクト

Project>New Project を選択して、新規プロジェクトのディレクトリを選択し、名前を入力します。ここでは MPLAB インストールディレクトリを使用し、名前を SAMPLE.PJT とします。

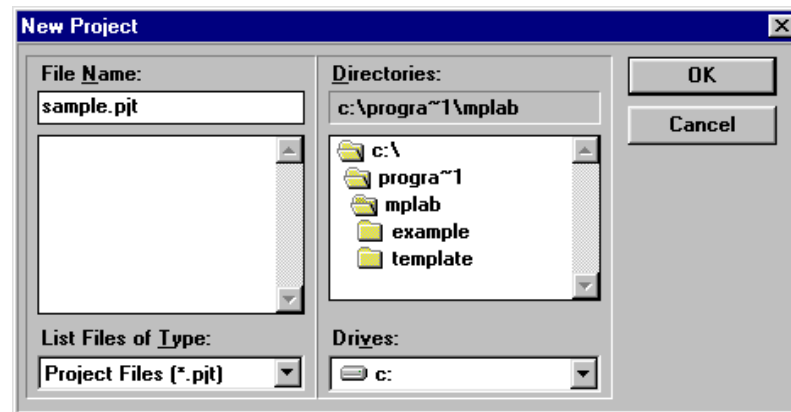


図 2

プロジェクトダイアログ

OK をクリックすると Edit Project ダイアログが開きます。

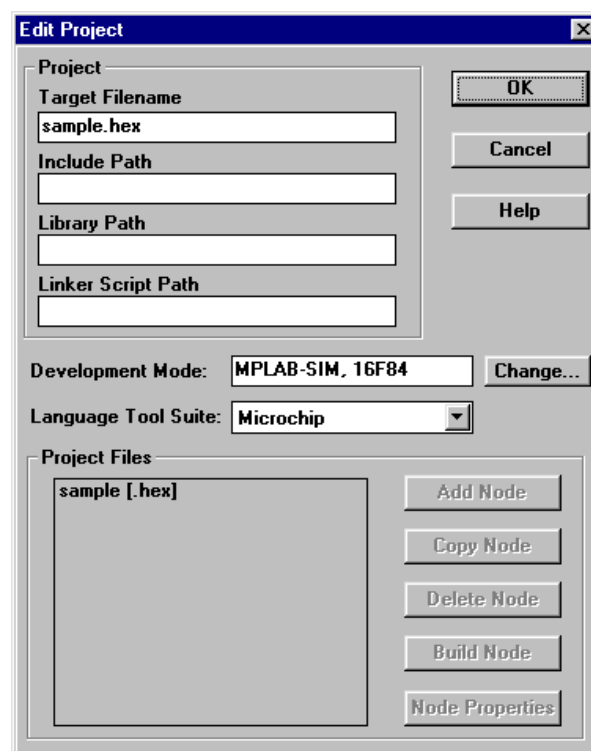


図 3

ノードプロパティの設定

「Project Files」ウィンドウで、ファイル名 SAMPLE.HEX をクリックし、「Node Properties」ボタンを選択します。

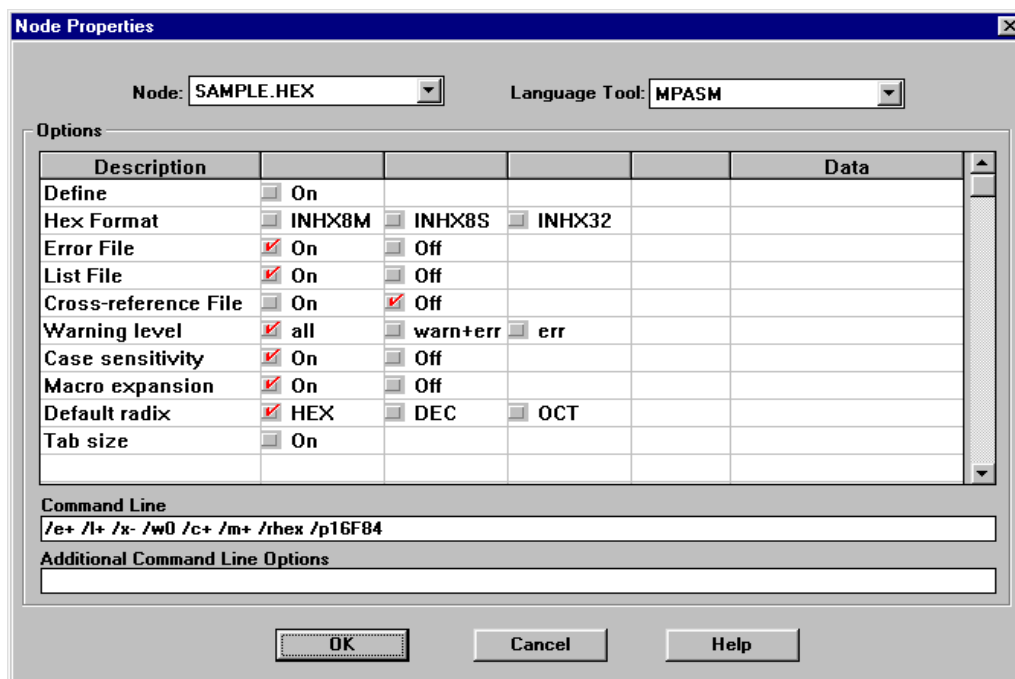


図 4

注意: 「Node Properties」ダイアログはツールのコマンドラインのスイッチ（この場合 MPASM）を表示します。最初にこのダイアログを開けると、チェックボックスにはこのツールのデフォルト値が表示されます。このチュートリアルではチェックボックスの変更を行う必要はありません。コマンドラインスイッチについての詳細は MPLINK、MPASM および MPLIB ユーザーズガイドを参照してください。

OK を選択すると、Edit Project ダイアログボックスに戻ります。

ノードの追加

Edit Project ダイアログから「Add Node」を選択します。ここでは SAMPLE.ASM を使用してください。これは、「Add Node」を選択すると現れるブラウザウィンドウです。

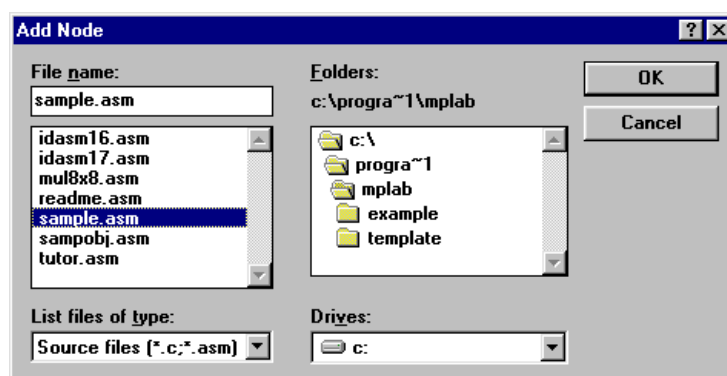


図 5

MPASM は常にソース.ASM ファイルと同じ名前の.HEX ファイルを作成します。プロジェクトが構築されると、プロジェクトマネージャーは SAMPLE.HEX ファイルを作成します。

Edit Project ダイアログは次のように表示されます。

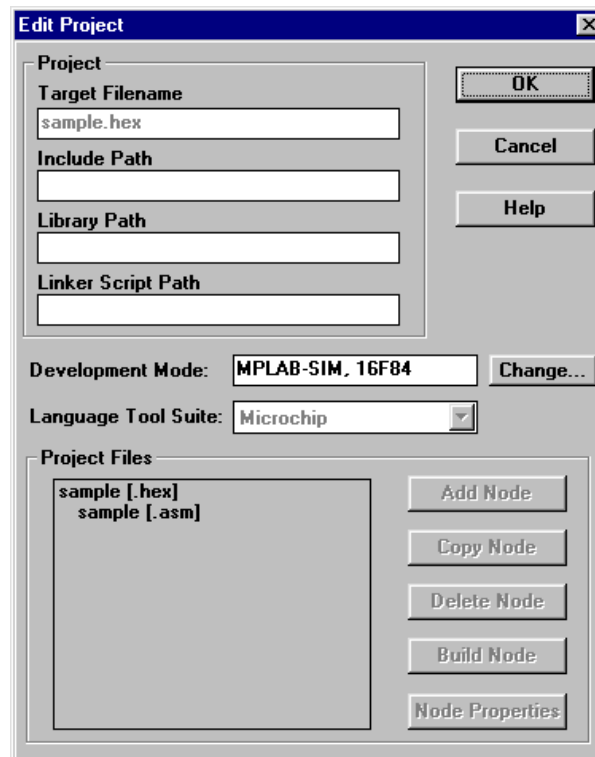


図 6

注意：この簡単な例では、「Path」ボックスの中に何も入力されていません。アプリケーションが複雑になるに従って、該当するボックスに Include File のディレクトリを入力する必要がある場合があります。

Edit Project ダイアログの「OK」ボタンをクリックします。

プロジェクトの作成

メニューから **Project>Make Project** を選択し、MPASM を使用してアプリケーションをコンパイルします。Build Result ウィンドウが作成され、アセンブラに送信されたコマンドラインを表示します。次のような表示になります。

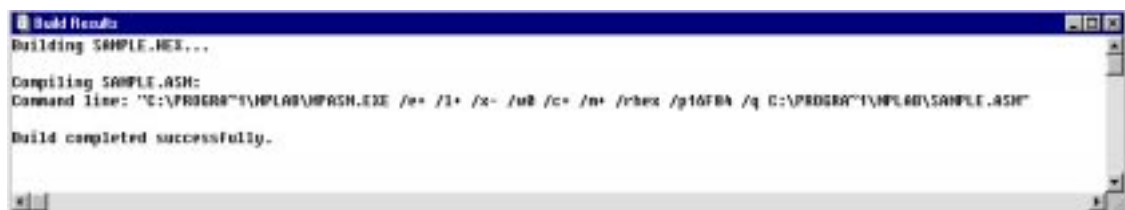


図 7

トラブルシューティング

問題がある場合は、次の項目をチェックします。

Project>Install Language Tool... を選択し、MPASM が MPLAB インストールディレクトリ内の MPASM.EXE に指定されていることを確認します。代わりに MPASM を MPASMWIN.EXE に指定することもできますが、「Windowed」オプションを選択する必要があります。

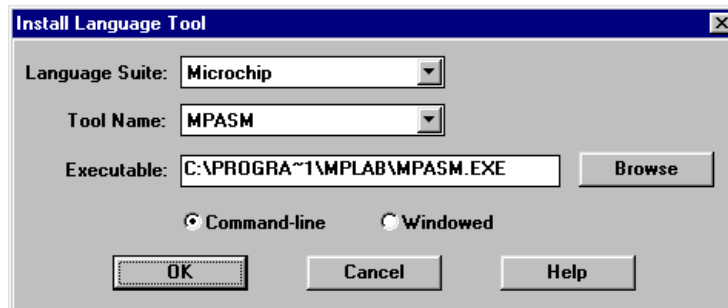


図 8

DOS から環境スペースが足りないというメッセージを受信した場合、MPLAB インストールディレクトリの MPASM.EXE ファイルを選択します。マウスの右ボタンをクリックして、プロパティダイアログを表示させます。



図 9

環境変数の初期サイズの容量を増やします。通常、2048 の設定で十分ですが、AUTOEXE.BAT ファイルの中に、変数を設定したり、パス文を追加するアプリケーションがたくさんある場合、環境変数の初期サイズを大きくする必要がある場合があります。

プロジェクトウィンドウ

Window>Project ウィンドウを開いて、ターゲット名がノードのソース名と一致するよう正しく設定されているのを確認します。ファイルの拡張子は、.ASM と .HEX のように異なりますが、このチュートリアルでは両方とも SAMPLE と名付けられています。

プロジェクトウィンドウは次のように表示されます。

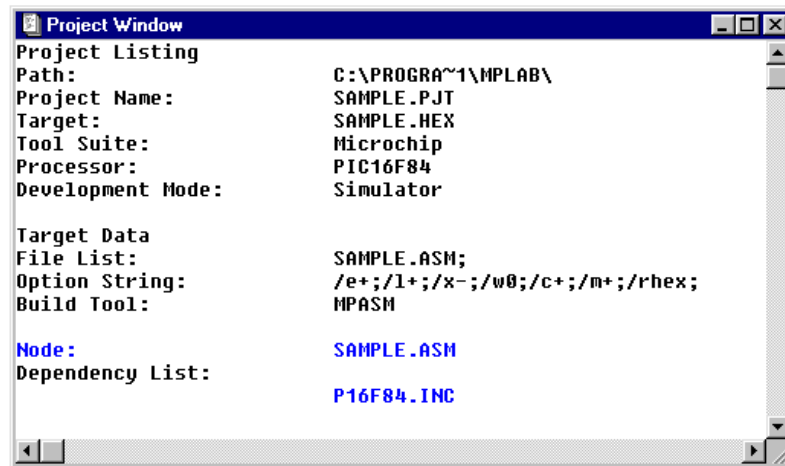


図 10

MPASM 1 つのファイルのプロジェクトのセットアップのまとめ

この章で説明した新規プロジェクトのセットアップ手順について簡単にまとめます。

- Project>NewProject で新規プロジェクトを作成します。
- プロジェクトのノードプロパティの設定を MPASM にして、必要なオプションを選択します。
- ソースファイルノードを追加します。

プロジェクト作成をしない単独の MPASM ソースファイルのコンパイル

プロジェクトを開かずに単独のファイルをコンパイルすることができます。最初のプロジェクトのセットアップは必要ありませんが、ファイルをコンパイルする時に毎回オプションを指定する必要があります。この例では、前の例と同じアセンブル言語ファイルを使用します。

まず、開いているプロジェクトをすべて閉じます。 Project>Close Project を選択します。

開発モードの設定

開発モードを選択します。ここでは、 Options>Development Mode の設定を MPLAB-SIM シミュレーターにして、PIC16F84 PICmicro を選択します。

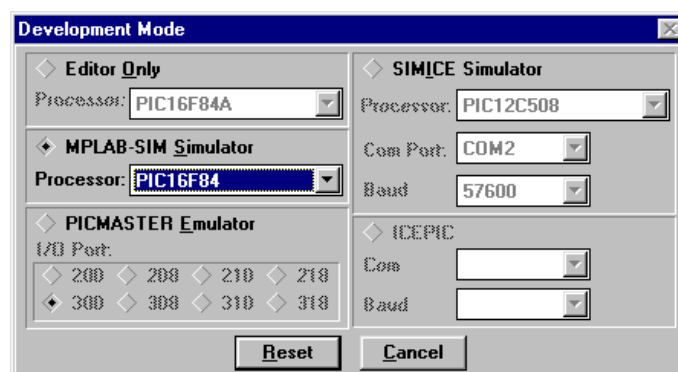


図 11

ソースファイルのオープン

アセンブルするソースファイルを開きます。ここでは MPLAB インストールディレクトリの 中に ある SAMPLE.ASM を使用します。

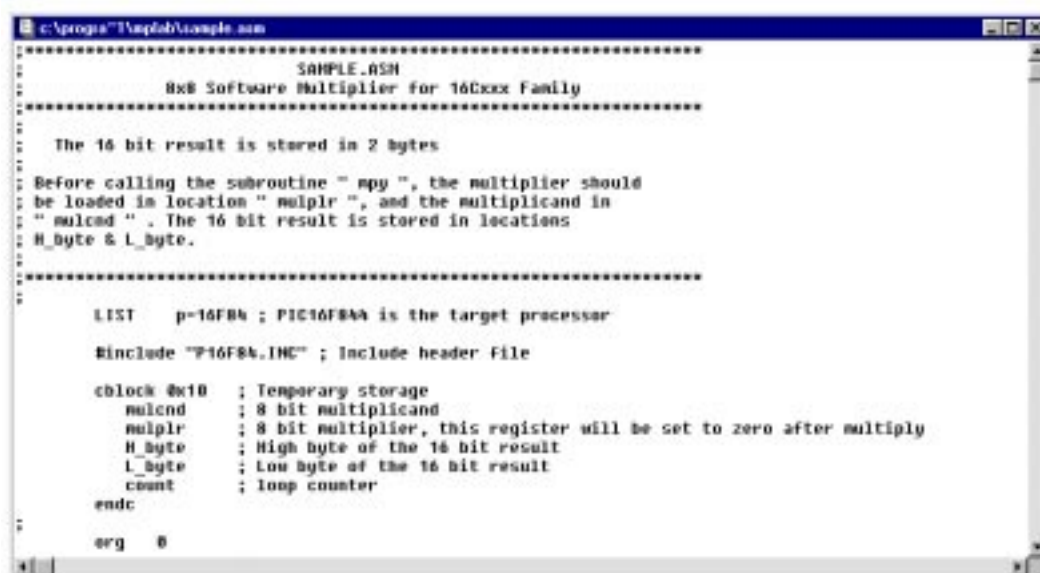


図 12

ソースファイルのコンパイル

メニューから **Project>Build Node** を選択し、MPASM を使用して SAMPLE.ASM をコンパイルします。次のような「Invoke Build Tool」ダイアログが現れます。

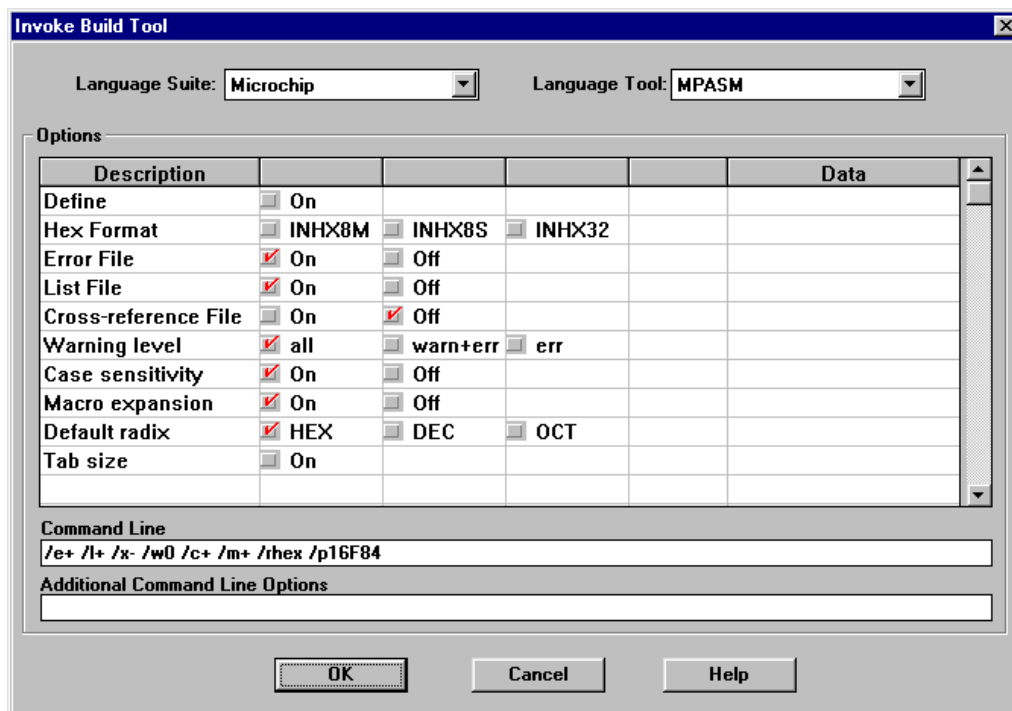


図 13

MPASM が選択されているのを確認し、ツールのオプションの設定を上記と一致させます。「Invoke Build Tool」ダイアログの「OK」をクリックしてビルドを開始します。「Build Results」ウィンドウが現われて、アセンブラーに送られたコマンドラインとビルド結果を表示します。次のような表示になります。

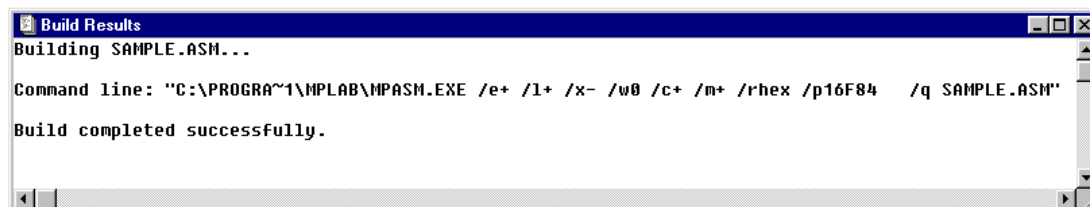


図 14

トラブルシューティング

問題がある場合は、次の項目をチェックします。

Project>Install Language Tool... を選択し、MPASM が MPLAB インストールディレクトリ内の MPASM.EXE に指定されていることを確認します。代わりに MPASM を MPASMWIN.EXE に指定することもできますが、「Windowed」オプションを選択する必要があります。

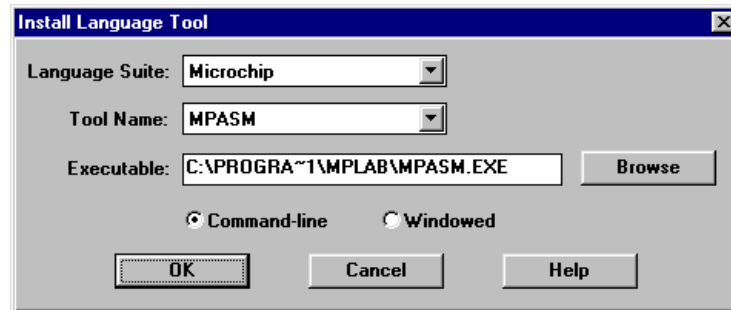


図 15

DOS から環境スペースが足りないというメッセージが表示された場合、Windows のエクスプローラを使用して、MPLAB インストールディレクトリ内の MPASM.EXE ファイルを選択します。次に、マウスの右ボタンをクリックして、プロパティダイアログを表示させます。



図 16

環境変数の初期サイズの容量を増やします。通常、2048 の設定で十分ですが、AUTOEXEC.BAT ファイルに変数を設定したり、パスステートメントを追加するアプリケーションが多くある場合、環境変数の初期サイズを大きくする必要があるかもしれません。

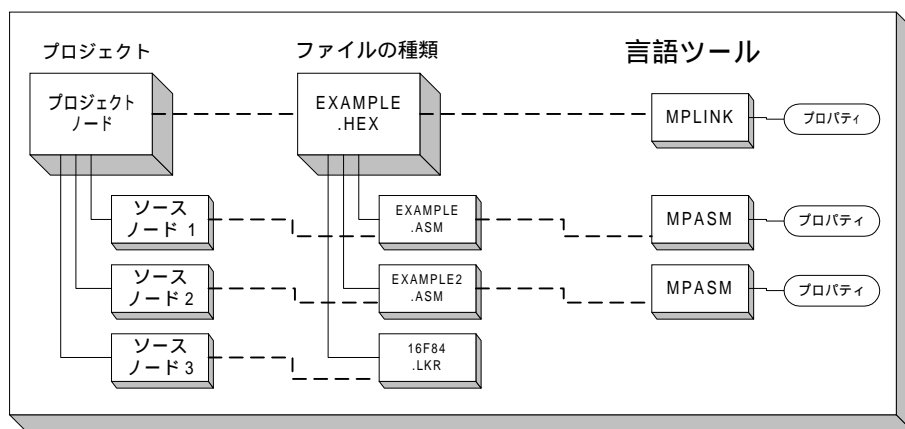
単独の MPASM ファイルのコンパイル設定のまとめ

この章で説明した新規プロジェクトのセットアップ手順について簡単にまとめます。

- オープン中のプロジェクトがあれば、Project>Close Project を使用してこれを閉じます。
- コンパイルしたいソースファイルを開きます。
- Project>Build Node を選択します。
- Invoke Build Tool ダイアログからの言語、ビルドツール、ビルドオプションを選択します。

複数の MPASM ソースファイルと MPLINK によるプロジェクトの作成

MPLINK を使用して複数のオブジェクトファイルをリンクするには、次の手順で行います。前章から続ける場合、Project>Close Project を選択します。



複数の MPASM ソースファイルのプロジェクト関連図

開発モードの設定

Options>Development Mode の設定を MPLAB-SIM シミュレータにして、この例では PIC16F84 を選択します。

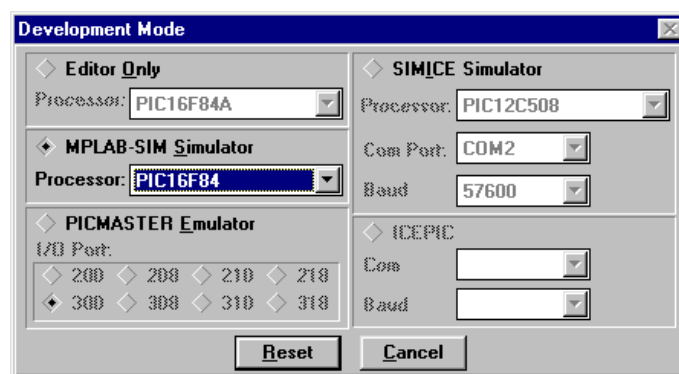


図 17

新規プロジェクト

Project>New Project を選択して、Browse により新規プロジェクトのディレクトリを選択し、名前を入力します。ここでは ¥MPLAB¥EXAMPLE を使用して EXAMPLE.PJT. と名付けてください。

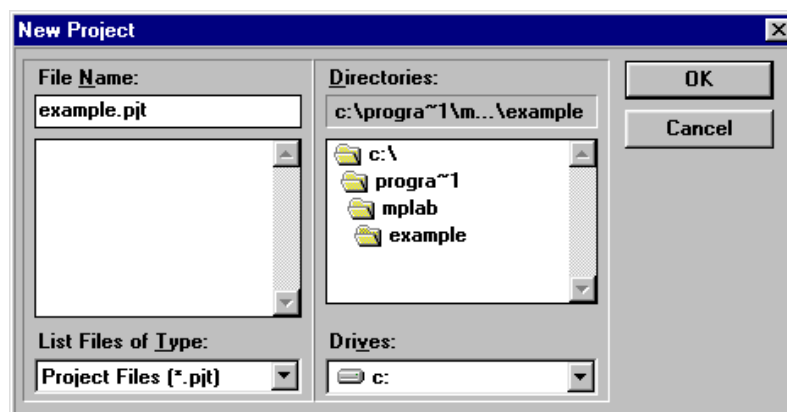
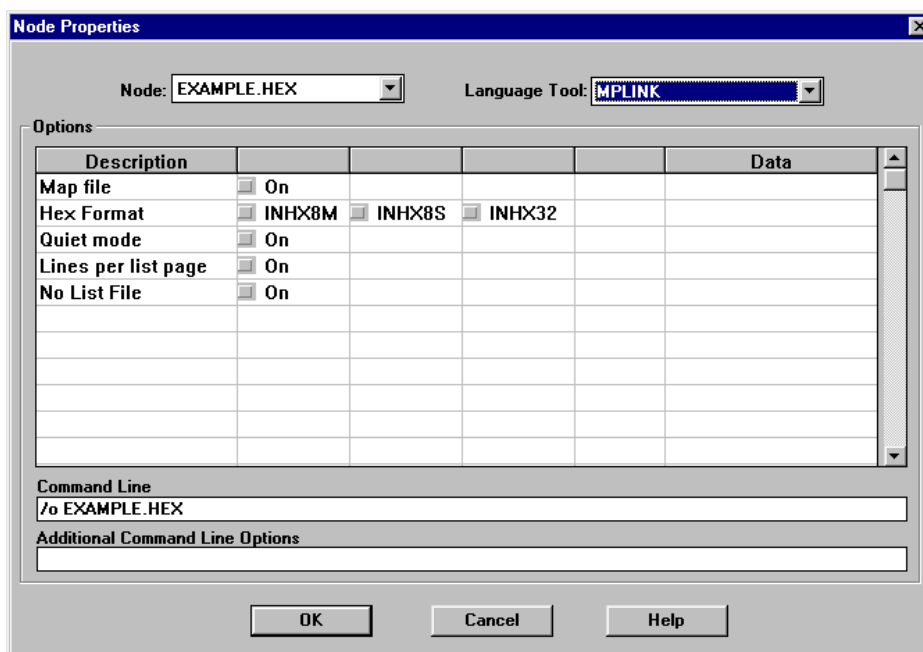


図 18

プロジェクトノードプロパティの設定



Edit Project ダイアログの「Project Files」ダイアログからプロジェクトノードのファイルネームを選択し、「Node Property」を押してこのダイアログを表示させます。言語ツールの設定を「MPLINK」にします。

図 19

注意: Node Property のダイアログはツールのコマンドラインスイッチ(この場合、MPLINK)を表示します。最初にこのダイアログを開くと、チェックボックスにはツールのデフォルト値が表示されます。このチュートリアルでは変更する必要はありません。これらのコマンドラインスイッチについての詳細は MPLINK、MPLIB および MPASM のユーザーズガイドを参照してください。

OK をクリックすると、Edit Project ダイアログボックスへ戻ります。

1 つめのソースファイルノードの追加

Edit Project ダイアログから「Add Node」を選択します。ここでは %MPLAB%EXAMPLE ディレクトリの EXAMPLE.ASM を使用します。

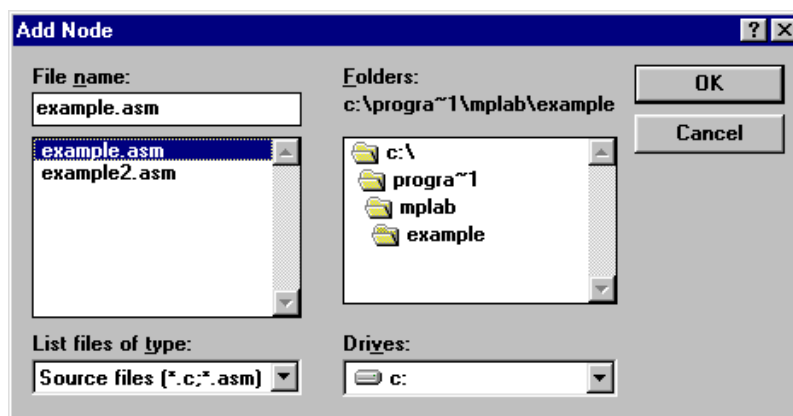


図 20

注意: Windows の標準的な方法を使用して、ダイアログから複数のファイルを一度に選択することもできます。この場合、1つのファイルを選択して、次にコントロールキー（2つのファイルのみ選択する）またはシフトキー（2つのファイルと間にあるすべてのファイルを選択する）を押しながら、もう1つのファイルを選択します。

Edit Project ダイアログ内のプロジェクトファイルリストから example.asm を選択し、「Node Properties」をクリックします。

言語ツールの設定が MPASM になっているか確認します。

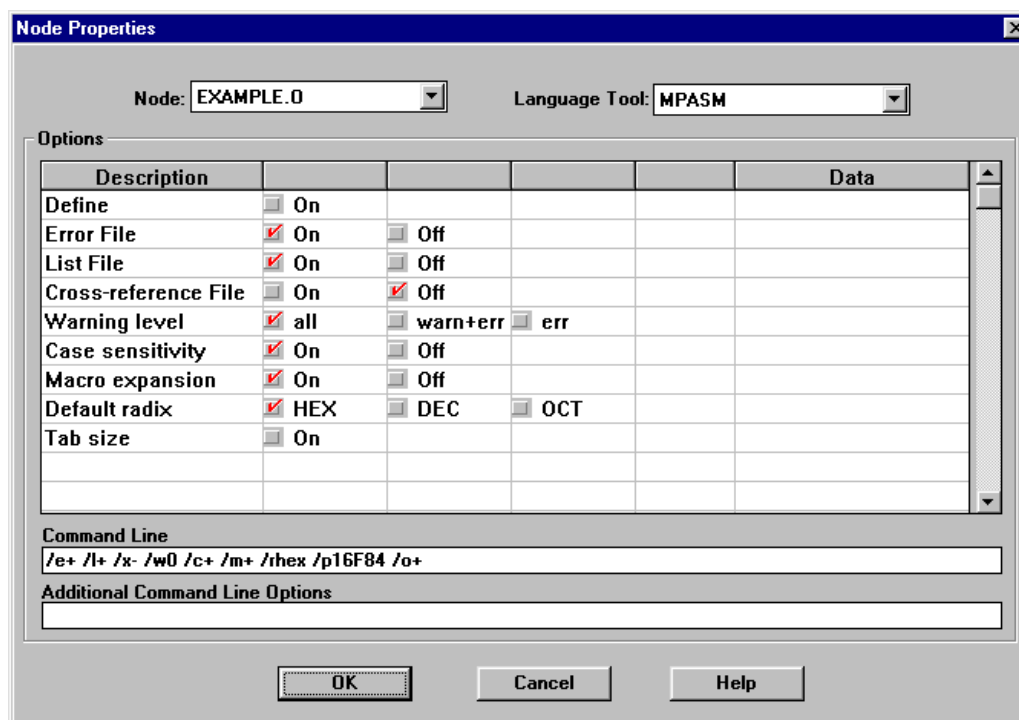


図 21

注意: Node Property ダイアログは、指定されたツールに対するコマンドラインスイッチ（この場合 MPASM）を表します。コマンドラインスイッチについての詳細は、MPLINK、MPLIB および MPASM ユーザーズガイドを参照してください。

OK を選択して、Edit Project ダイアログボックスへ戻ります。

ソースファイルの追加

前の2ステップと同様に、残りのソースファイルをプロジェクトに追加します。ここでは¥MPLAB¥EXAMPLE ディレクトリから EXAMPLE2.ASM を選択します。「Copy Node」オプションを使用して、1つめのソースファイルと同じ Node Option の設定で継続してファイルを入力することもできます。Node Option が各ファイルに正しく設定されていることを確認してください。

注意：「Copy Node」を使用するには、Project Files ボックス内にリストされたソースノードファイルを1つ選択してください。次に「Copy Node」機能を選択します。「Add Node」ダイアログボックス内のソースファイルを1つまたは複数選択します。ファイルを選択したら、OK ボタンを選びます。この手順では選択したファイルは、「Copy Node」を選択した時に選択されていた参照ノードと同じノードプロパティをセットアップすることになります。これは同じノードプロパティを持つ複数のソースファイルを追加する時に便利です。

リンカスクリプトの選択

「Add Node」ボタンと前出の方法を使用して、リンカスクリプトを選択します。リンカスクリプトは MPLINK が使用するファイルで、各 PICmicro のメモリアーキテクチャーを定義します。標準のリンカスクリプトは MPLINK と共に MPLAB インストールディレクトリ内にあります。ここでは¥MPLAB¥EXAMPLE ディレクトリから 16F84.LKR を選択します。ノードオプションはリンカスクリプトには設定できません。

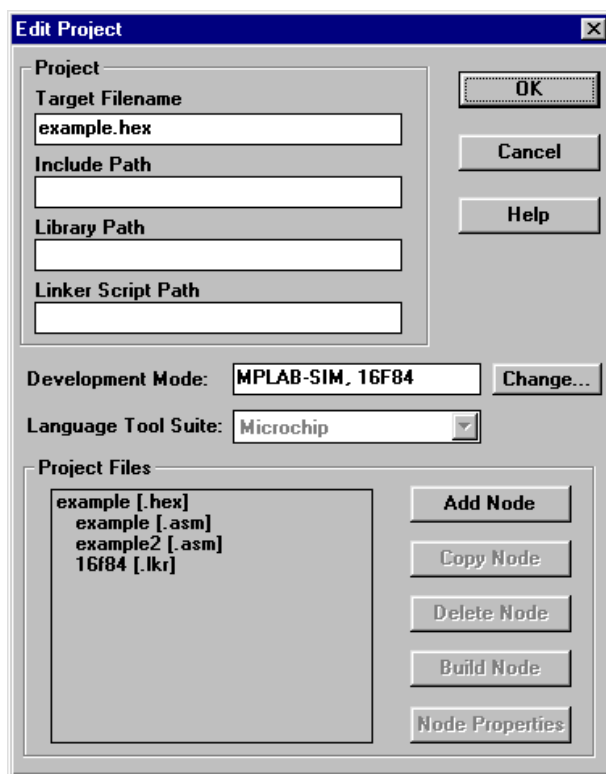


図 22

Edit Project ダイアログの OK をクリックします。

注意:この簡単な例では、3つの「Path」ボックスには何も入力されていません。アプリケーションが複雑になるに従って、適切なボックスに Include ファイル、Library ファイルおよび Linker Script ファイルのディレクトリを入力する必要がある場合があります。(Include ファイル、Library ファイル、Linker Script ファイルがプロジェクトと同じディレクトリにない場合)

プロジェクトの作成

メニューから **Project>Make Project** を選択し、MPASM と MPLINK を使用してアプリケーションをコンパイルします。「Build Results」ウィンドウが開いて、各ツールへ送信されたコマンドラインを表示します。次のように表示されます。

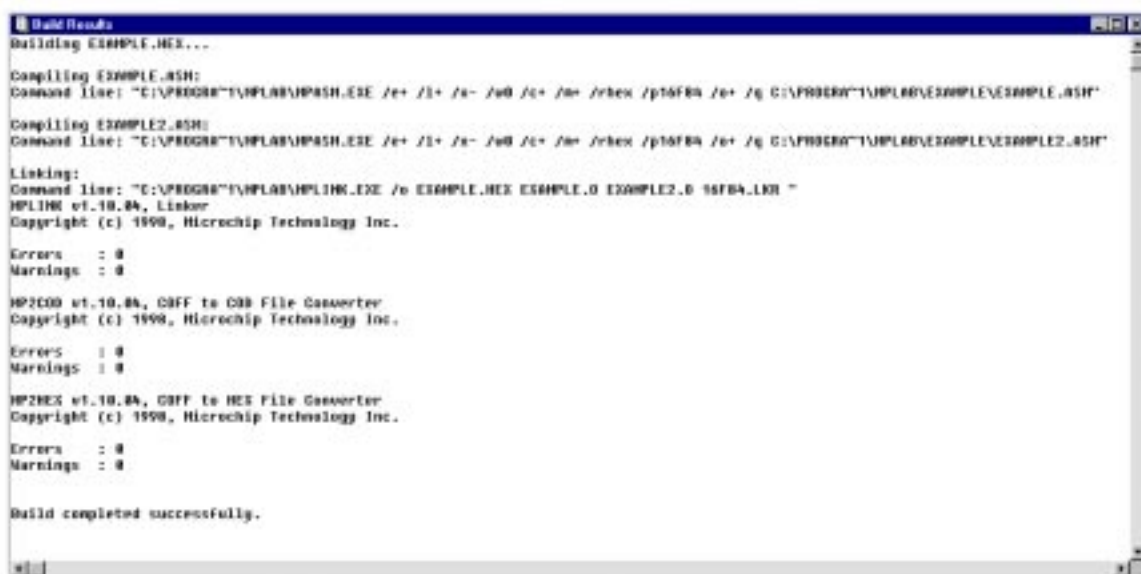


図 23

トラブルシューティング

うまく動作しない場合は、次の項目をチェックします。

MPLAB が「Time-out」というメッセージを表示した場合、OK を選択して続けます。PC の処理スピードやプロジェクトのサイズに応じて、MPLAB がタイムアウトのメッセージを発するまでの時間を調整したい場合もあります。この時間は、**Options>Environment Setup...** を選択し、ダイアログボックス内のタイムアウト設定を調整することで設定します。

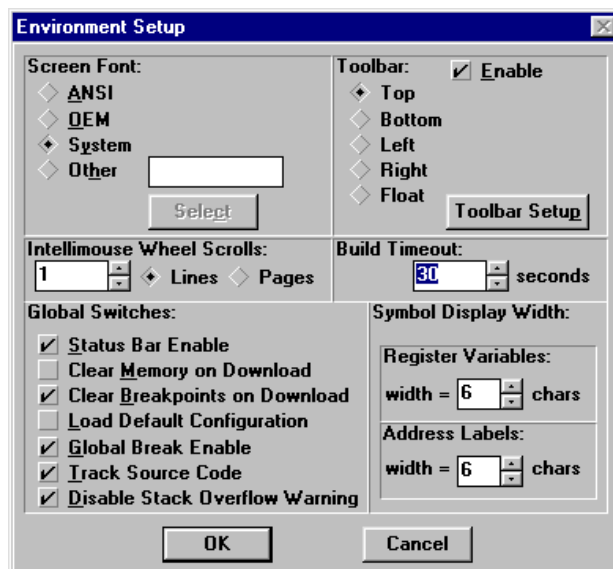


図 24

Project>Install Language Tool... を選択し、MPASM と MPLINK が MPLAB インストールディレクトリで MPASM.EXE および MPLINK.EXE を示しているかチェックします。

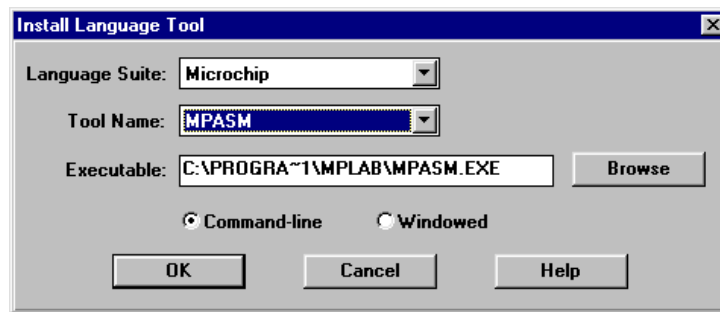


図 25

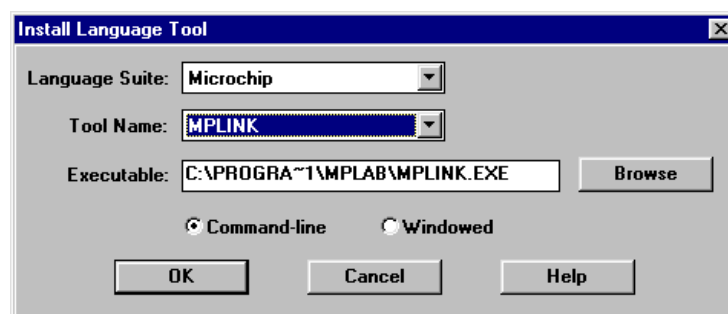


図 26

DOS から環境スペースが不足しているというメッセージを受信した場合、Windows のエクスプローラを使用して MPLAB インストールディレクトリの MPASM.EXE ファイルを選択します。次にマウスの右ボタンをクリックして、プロパティダイアログを開きます。



図 27

環境変数の初期サイズの容量を増やします。通常、2048 の設定で十分ですが、AUTOEXEC.BAT ファイルに変数を設定したり、パス文を追加するアプリケーションが多くある場合、環境変数の初期サイズを大きくしなければならない場合があります。

プロジェクトウィンドウ

Window>Project ウィンドウを開きます。次のように表示されます。

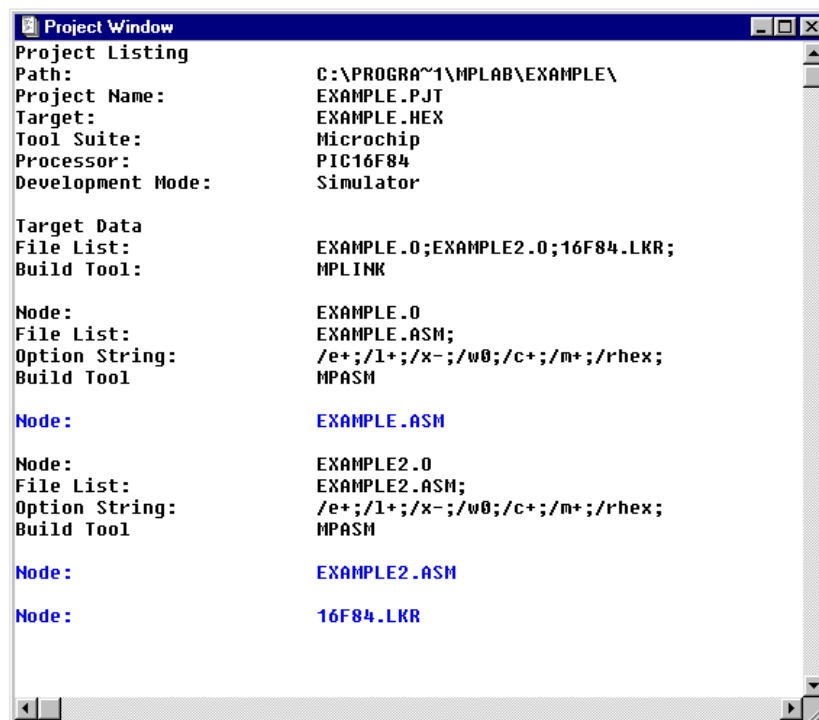


図 28

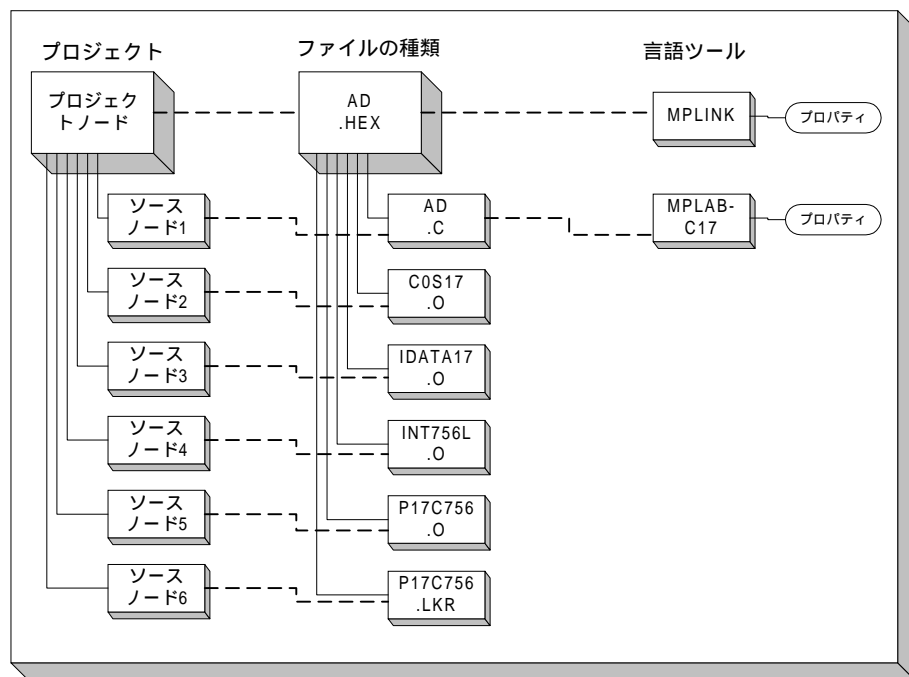
MPLINK による MPASM 複数ファイルプロジェクトのまとめ

この章で説明した新規プロジェクトのセットアップ手順について簡単にまとめます。

- Project>NewProject により新規のプロジェクトを作成します。
- プロジェクト Node プロパティの設定を MPLINK にします。
- ソースファイルノードを追加して、必要に応じて、ノードプロパティを設定します。
- Linker Script ファイルノードを追加します。

MPLAB-C17 によるプロジェクトの作成

このチュートリアルでは MPLAB のプロジェクト上で MPLAB-C17 を使用してアプリケーションを構築する方法を示します。前章から続けている場合は、Project>Close Project を選択します。



MPLAB-C17 チュートリアルプロジェクト関連図

開発モードの設定

Options>Development Mode を MPLAB-SIM シミュレータにセットして、デバイスを選択します。以下の例では 17C756 を選択しています。

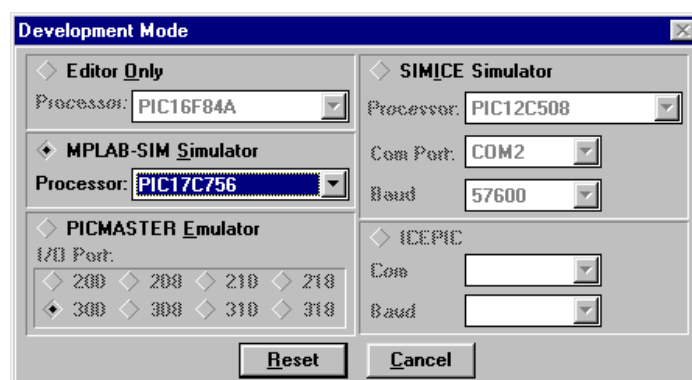


図 29

MPLAB-C17 言語ツールのインストール

MPLAB-C17 が MPLAB に正しくインストールされていることを確認します。Project>Install Language Tool ダイアログが次のように表示されます。(Executable のパスは異なる場合があります。)

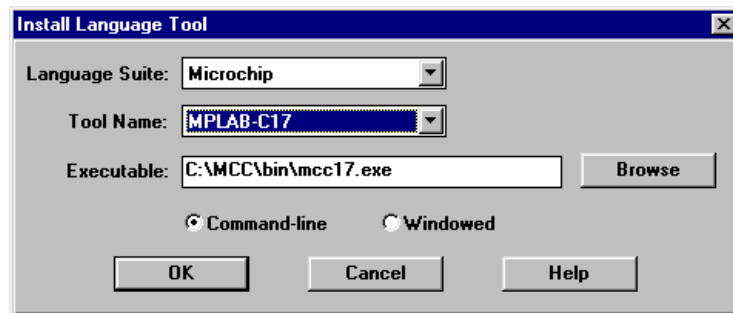


図 30

ウィンドウに Executable の表示が出ない場合は、Browse ボタンを使用して MCC17.EXE を探して選択します。

新規のプロジェクト

Project>New Project を選択して、新規プロジェクトのディレクトリを選択し、名前を入力します。¥MCC¥EXAMPLES¥AD ディレクトリ内で、AD.PJT と名付けます。

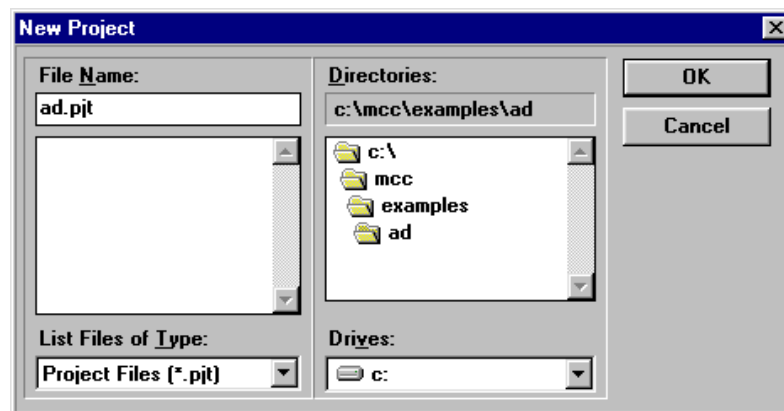


図 31

プロジェクト名の設定後、OK をクリックすると、Edit Project ダイアログが表示されます。

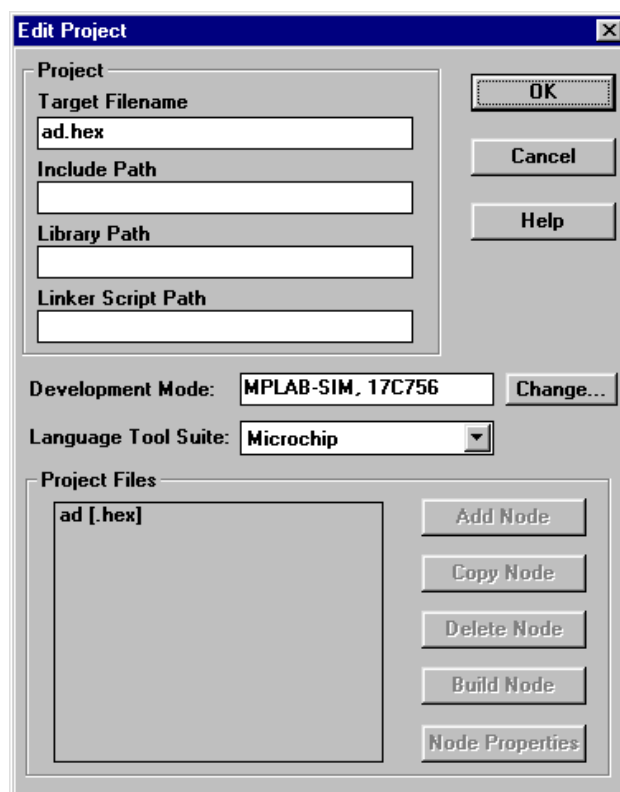


図 32

プロジェクトノードに対するプロジェクトオプションの設定

Edit Project ダイアログの「Project Files」からプロジェクトファイルの名前を選択して、「Node Properties」をクリックします。言語ツールの設定を「MPLINK」にします。

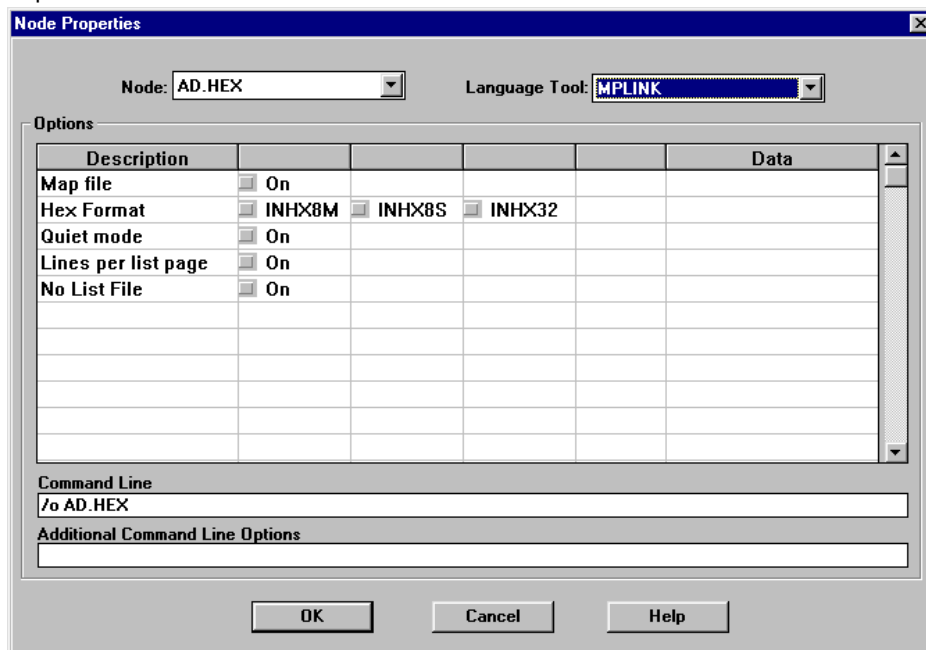
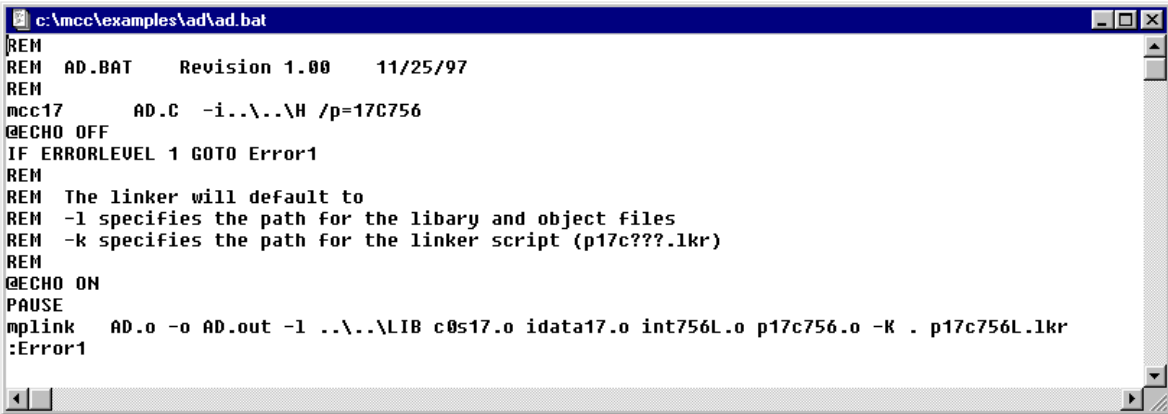


図 33

注意: Node Properties ダイアログは、指定したツールのコマンドラインスイッチ（この場合 MPLINK）を示します。このダイアログを開くのが1回目の時は、チェックボックスにはそのツールのデフォルト値が表示されます。このチュートリアルではチェックボックスを変更する必要はありません。このコマンドラインスイッチについての詳細は、MPLINK による MPASM および MPLIB ユーザーガイドを参照してください。

1 つめのソースファイルの追加

このチュートリアルでどのようにノードをセットアップしたらよいかについては、AD.BAT を参考にしましょう。これは、DOS コマンドラインでこのチュートリアルをコンパイルするためのバッチファイルで、¥MCC¥EXAMPLES¥AD ディレクトリ内にあります。このバッチファイルの内容を参照してすべての必要なノードを追加します。以下はバッチファイルのリストです。



```
c:\mcc\examples\ad\ad.bat
REM
REM AD.BAT      Revision 1.00    11/25/97
REM
mcc17      AD.C  -i..\..\H /p=17C756
@ECHO OFF
IF ERRORLEVEL 1 GOTO Error1
REM
REM The linker will default to
REM -l specifies the path for the library and object files
REM -k specifies the path for the linker script (p17c???.lkr)
REM
@ECHO ON
PAUSE
mplink  AD.o -o AD.out -l ..\..\LIB c0s17.o idata17.o int756L.o p17c756.o -K . p17c756L.lkr
:Error1
```

図 34

必要なノードは、AD.C（必ずコンパイルするメインソースファイル）、リンクスクリプト P17C756L.LKR、および次のオブジェクトファイルです。

- | | |
|-----------|---------------------|
| C0S17.0 | - スタートアップコード |
| IDATA17.0 | - データ初期化用のコード |
| INT756L.0 | - 割り込み処理ルーチン |
| P17C756.0 | - PIC17C756 レジスタの定義 |

Project>Edit Project メニューを選択すると、プロジェクトのセットアップに戻ることができます。

Edit Project ダイアログから「Add Node」を選択します。 ¥MCC¥EXAMPLES¥AD ディレクトリからソースファイル AD.C を追加します。

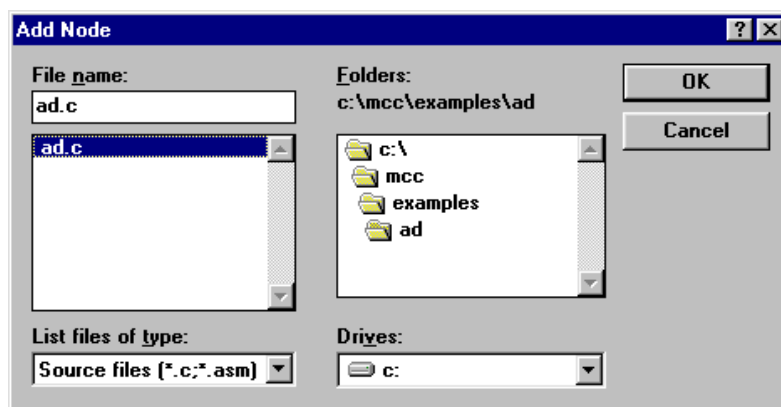


図 35

Edit Project ダイアログ内のファイル名を選択して「Node Properties」をクリックします。言語ツールの設定が MPLAB-C17 になっているのを確認します。

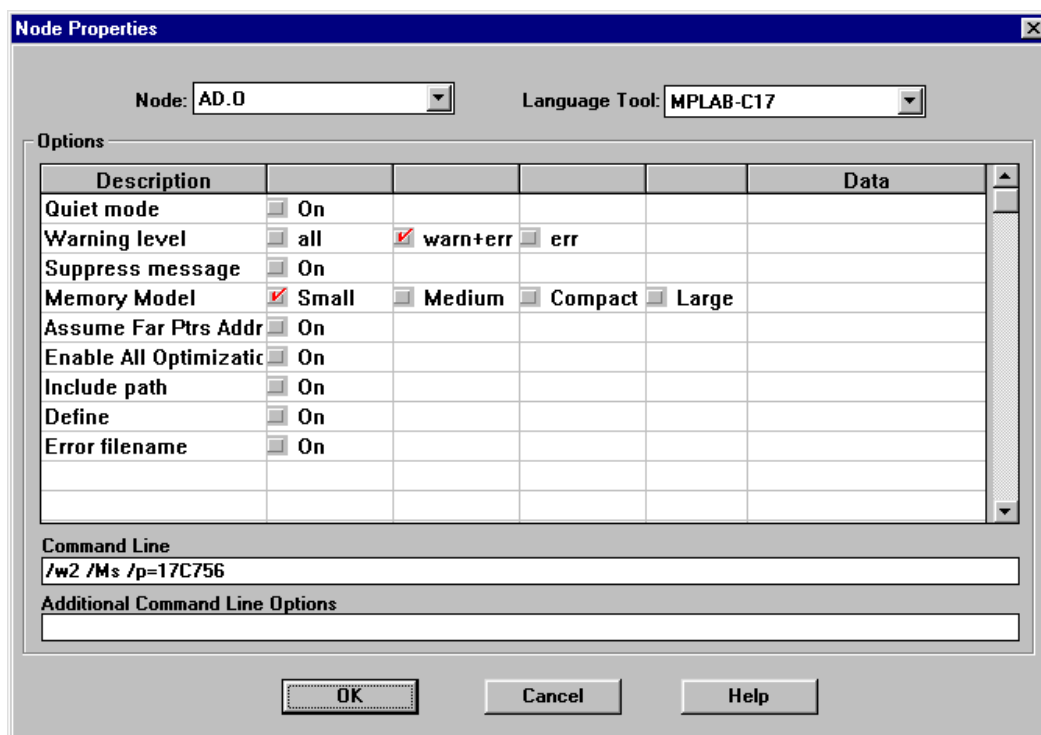


図 36

「Object ファイル名」は自動的に「AD.0」と設定されます。このダイアログでは他に何も変更する必要がありません。

注意: Node Properties ダイアログは、指定したツール（この場合は MPLAB-C17）のコマンドラインスイッチを示します。チェックボックスはそのツールのデフォルト値を表示します。このチュートリアルでは、デフォルト値のままでどの設定も変更する必要はありません。コマンドラインスイッチについての詳細は、MPLAB-C17 ユーザーズガイドを参照してください。

コンパイル済みオブジェクトファイルの追加

Edit Project ダイアログから「Add Node」ボタンを使用して、¥MCC¥LIB 内の MPLAB-C17 ライブラリからコンパイル済みのオブジェクトファイルを追加します。必要な全オブジェクトファイル、COS17.0、IDATA17.0、INT756L.0 および P17C756.0 を追加します。複数のファイルを一度に選択するには、キーボードの「Ctrl」キーを押しながら、マウスでファイルを選択します。コンパイル済みのオブジェクトファイルにはオプションを設定できません。

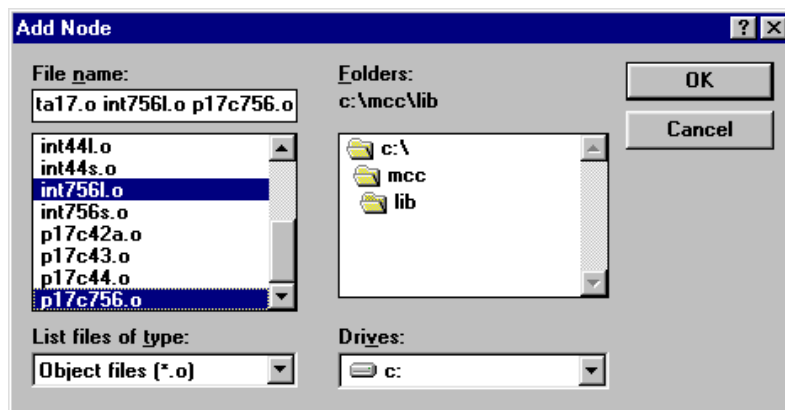


図 37

xxx.lib（ライブラリ）ファイルを追加するには、上記と同じ手順で行えます。

リンカスクリプトの選択

リンカスクリプトを選択して、ノードとして追加します。¥MCC¥EXAMPLES¥AD ディレクトリのリンカスクリプト、P17C756L.LKR を使用します。リンカスクリプトにはオプションの設定ができません。

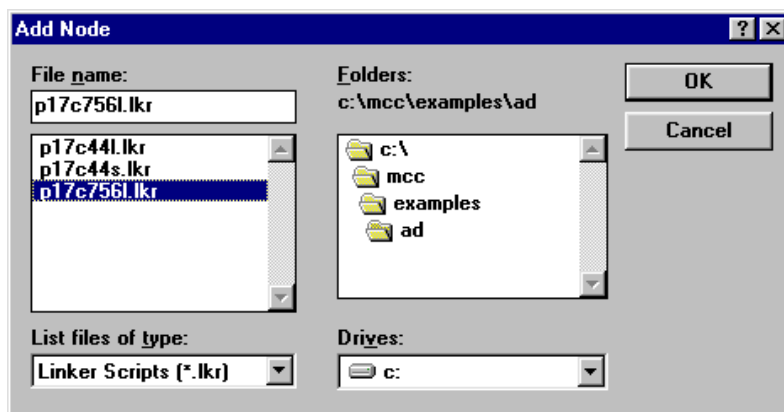


図 38

Edit Project ウィンドウは次のように表示されます。

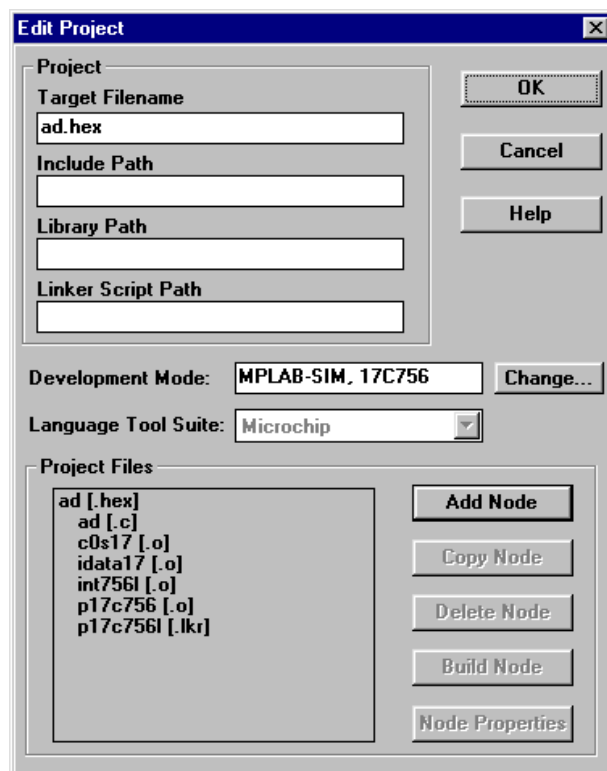


図 39

Edit Project ダイアログ上で、「OK」をクリックします。

プロジェクトの作成

メニューから **Project>Make Project** を選択して、MPLAB-C17 と MPLINK を使用したアプリケーションをコンパイルします。Build Result ウィンドウが作成されて、各ツールを実行するコマンドラインを表示します。次のような表示になります。



図 40

トラブルシューティング

問題がある場合は、次の項目をチェックします。

Project>Install Language Tool... を選択して、MPLAB-C17 と MPLINK の表示が、MCC17.EXE と MPLINK.EXE になっているか確認します。

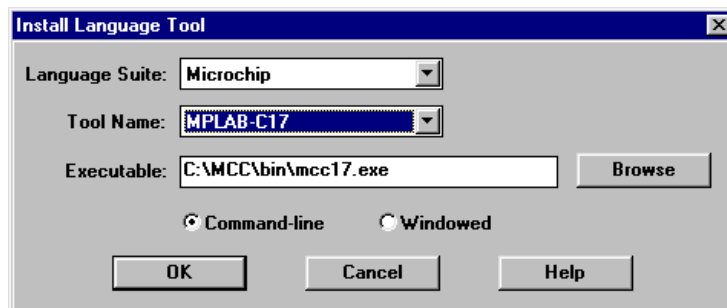


図 41

プロジェクトウィンドウ

Window>Project ウィンドウを開くと、次のように表示されます。

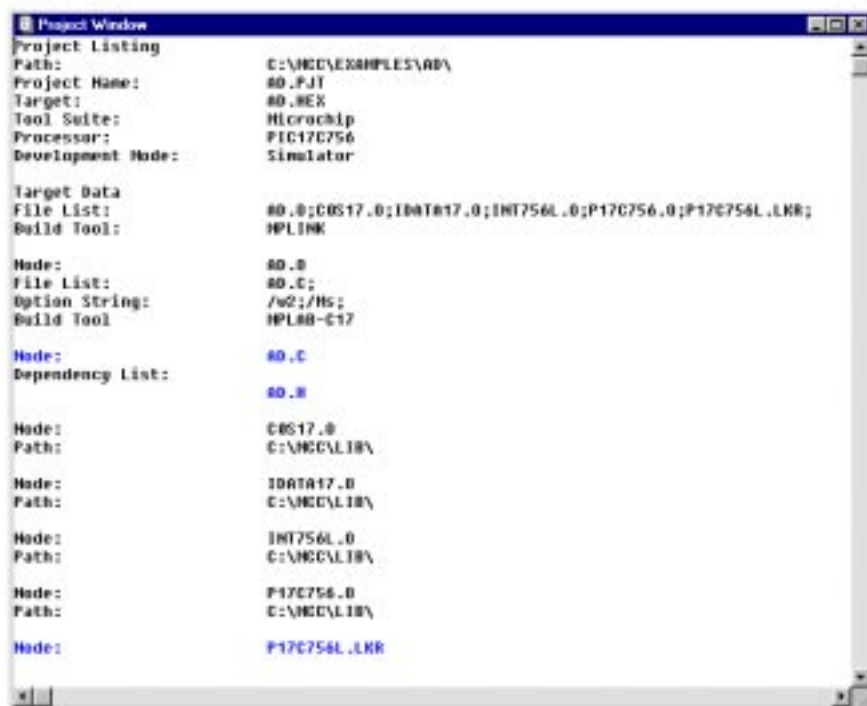


図 42

MPLAB-C17 プロジェクトのセットアップ方法のまとめ

この章で説明した新規プロジェクトのセットアップ手順について、簡単にまとめます。

- Project>NewProject で新規のプロジェクトを作成します。
- プロジェクトノードのプロパティの設定を MPLINK にします。
- ソースファイルを追加して言語ツールの設定を MPLAB-C17 または MPASM にします。
- コンパイル済みのノードを追加します。(.0 ファイルおよび .LIB ファイル)
- リンカスクリプトノードを追加します。

Hi-Tech PIC C によるプロジェクトの作成

この章では、MPLAB のプロジェクトで Hi-Tech's PIC C コンパイラを使用してアプリケーションを構築する方法を示します。前章から続けている場合は、Project>Close Project を選択します。

開発モードの設定

Options>Development Mode の設定を MPLAB-SIM シミュレータにして、品名を選択します。この例では 16C77 PICmicro を選択しています。

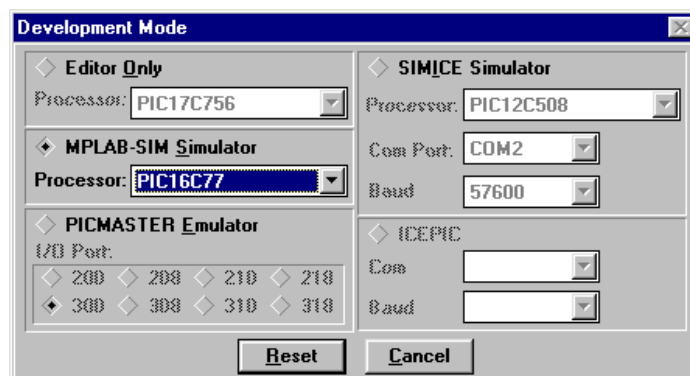


図 43

PIC C 言語ツールのインストール

次の様に PIC C を MPLAB にインストールします。Project>Install Language Tool ダイアログで HI-TECH ツールを次の例のように設定する必要があります。(パスは異なる場合があります。)

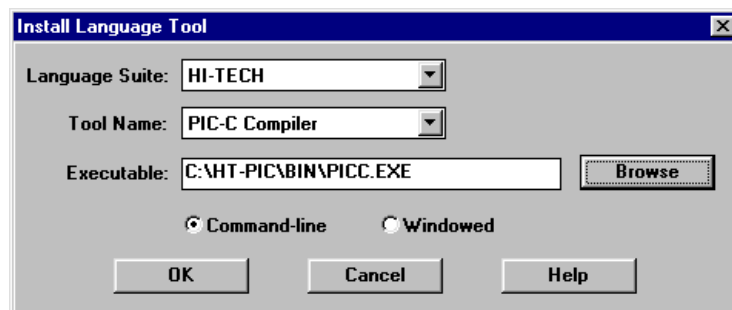


図 44

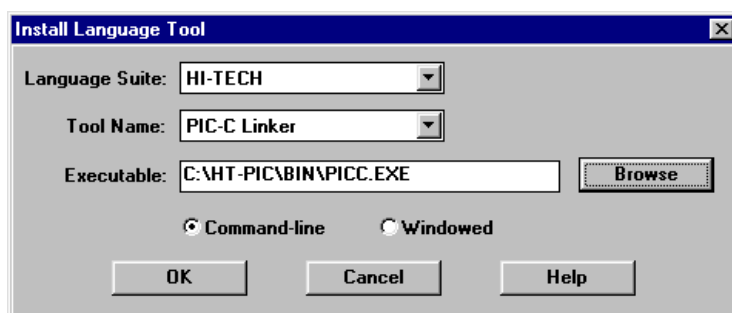


図 45

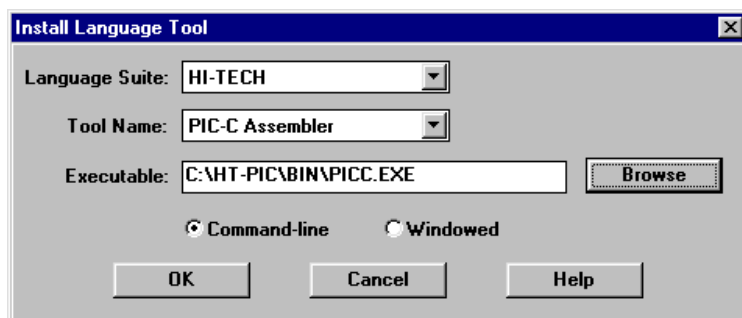


図 46

Browse ボタンを使用して、コンパイラ、リンカ、アセンブラとして PICC.EXE を 指定します。

新規のプロジェクト

Project>New Project を選択して、新規プロジェクト用のディレクトリを選択後、名前を入力します。¥HT-PIC¥SAMPLES ディレクトリ内で、SAMPLE.PJT と名付けます。

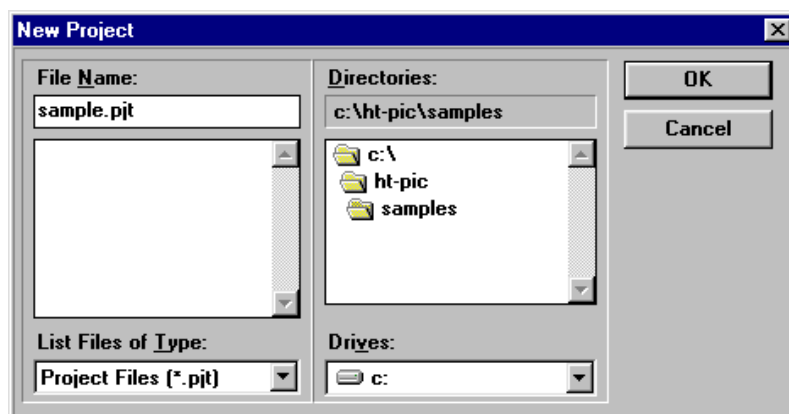


図 47

プロジェクト名の設定後、OK をクリックすると、Edit Project ダイアログが次のように表示されます。

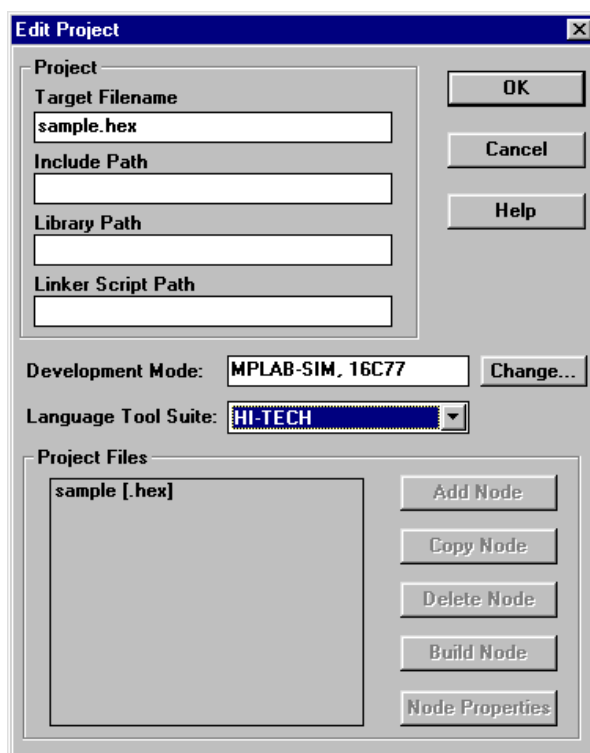


図 48

Language Tool Suite の設定を HI-TECH にします。

プロジェクトオプションの設定

Edit Project ダイアログボックスの「Project Files」ダイアログでプロジェクトの名前を選択して、「Node Properties」をクリックします。次に、言語ツールを「PIC-C Linker」に設定して、「Generate Debug Info」のチェックボックスをオンにします。最後に、次の例のように、Data 列に「Microchip」と入力します。

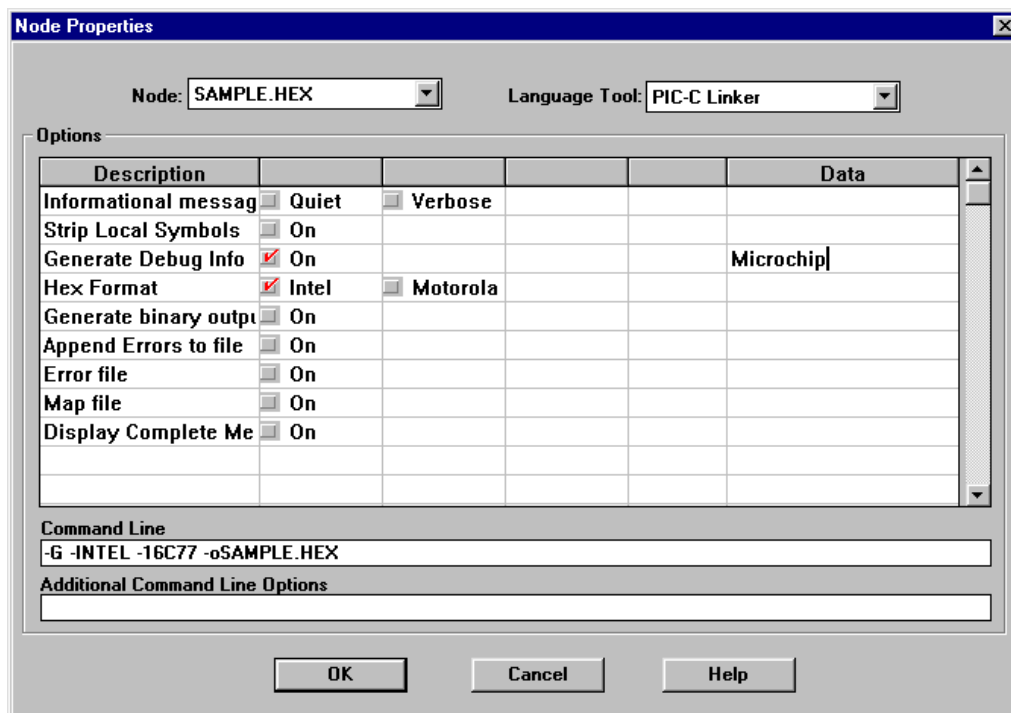


図 49

注意: Node Properties ダイアログはツール（この場合 PIC-C）のコマンドラインスイッチを示します。このダイアログボックスを開くのが 1 回目の場合、チェックボックスにはツールに対するデフォルト値が表示されます。このチュートリアルで変更が必要なのは、デバッグ情報のセッティングだけです。このコマンドラインスイッチについての詳細は、Hi-Tech のドキュメントを参照してください。

Node Properties ダイアログで「OK」をクリックして、Edit Project ダイアログに戻ります。

1 つめのソースファイルの追加

Edit Project ダイアログから「Add Node」を選択します。¥HT-PIC¥SAMPLES ディレクトリにある SAMPLE.C というソースファイルを追加します。

Add Node ダイアログでファイル名が表示されたら、そのファイル名を選択し、「Node Properties」をクリックします。

このダイアログは以下のように設定します。

- 「Language Tool」の設定を PIC-C コンパイラにします。
- 「Generate Debug Info」のチェックボックスをオンにします。
- 「Generate Debug Info」の Data 列に「Microchip」と入力します。

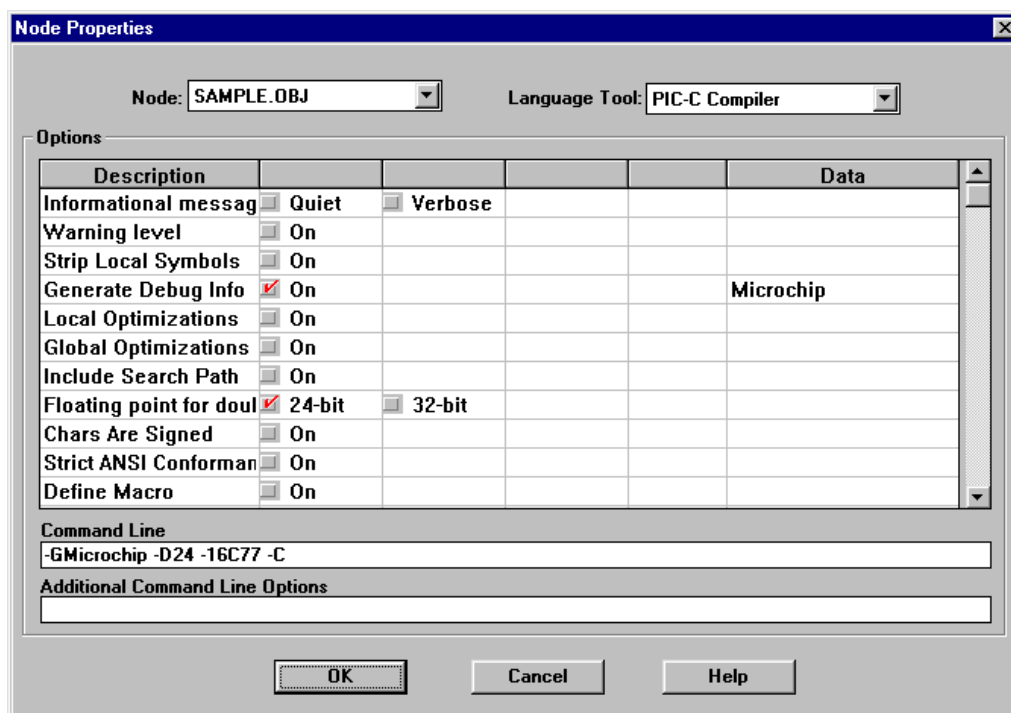


図 50

「Object filename」は自動的に「SAMPLE.OBJ」と設定されます。

注意： Node Properties ダイアログはツール（この場合は PIC C）のコマンドラインスイッチを示します。チェックボックスにはツールのデフォルト値が表示されます。ここで変更の必要があるのは、デバッグ情報だけです。コマンドラインスイッチについての詳細は、Hi-Tech PIC C のドキュメントを参照してください。

OK をクリックして、SAMPLE.C ノードを選択します。次に、「Copy Node」ボタンを使用して、SAMPLE.C と同じ Node Properties に ADC.C、DELAY.C、LCD.C を追加します。その後、プロジェクトの表示は次のようになります。

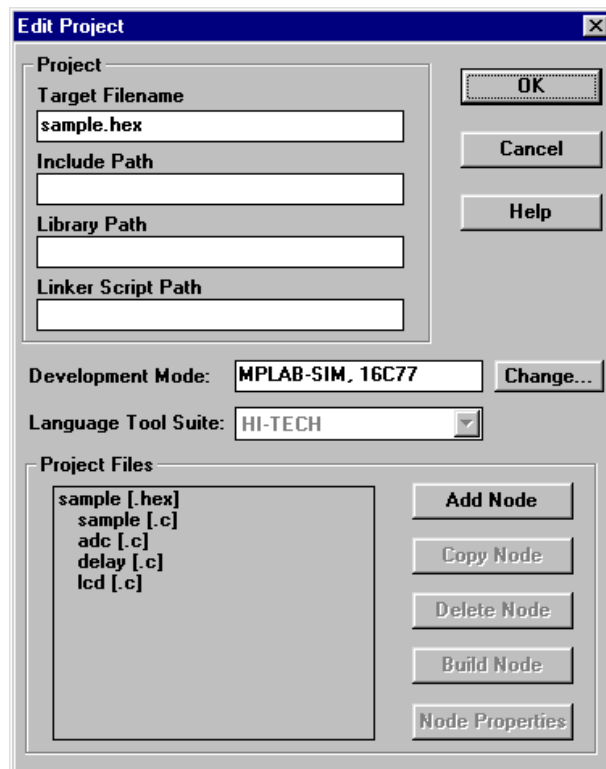


図 51

プロジェクトの作成

メニューから **Project>Make Project** を選択し、Hi-Tech コンパイラとリンカを使用してアプリケーションをコンパイルします。各ツールを実行する際のコマンドラインを示す Build Results ウィンドウが作成されます。このウィンドウは次のように表示されます。

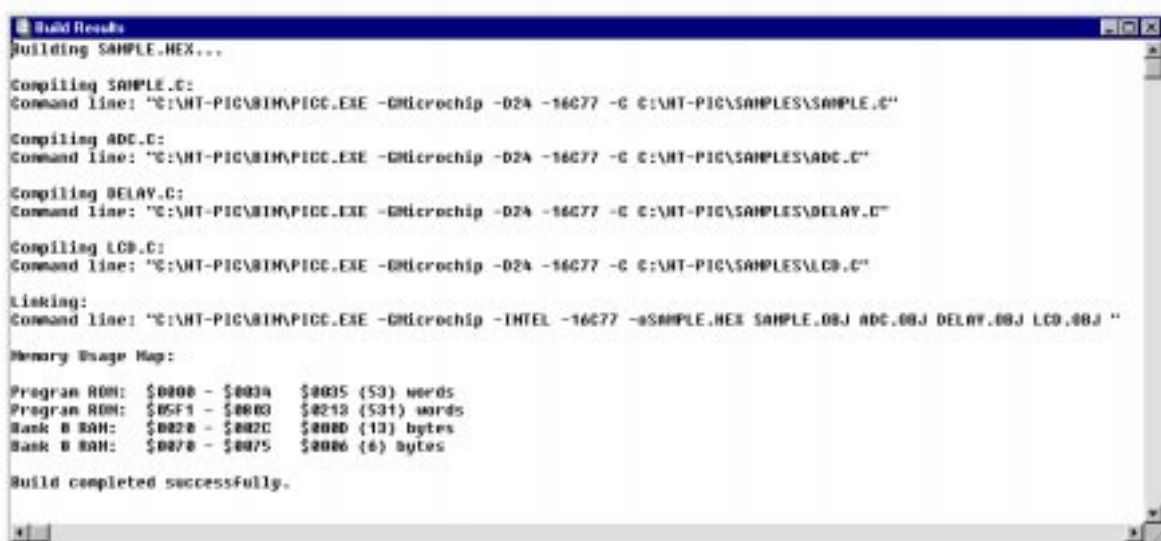


図 52

トラブルシューティング

問題がある場合は、以下の項目をチェックします。

Project>Install Language Tool... を選択して、PIC C コンパイラと PIC C リンカがどちらも PICC.EXE になっていることを確認します。

プロジェクトウィンドウ

Window>Project ウィンドウを開くと、次のように表示されます。



図 53

PIC C プロジェクトセットアップのまとめ

この章で説明した新規プロジェクトのセットアップ手順について簡単にまとめます。

- PIC C コンパイラ、リンカ、アセンブラの言語ツールをセットアップします。
- 「Include File Directory」を¥HT-PIC¥H (または PIC C の Include ファイルがインストールされている場所) にセットする必要がある場合があります。
- Project>NewProject で新規プロジェクトを作成します。
- プロジェクトノードの「Generate Debug Info」をオンにします。
- プロジェクトノードのプロパティの設定を PIC C リンカにします。
- ソースファイルを追加し、言語ツールの設定を PIC C コンパイラまたはアセンブラにします。
- 各ソースノードの「Generate Debug Info」をオンにします。
- 各ソースノードの「Generate Debug Info」の Data 列を「Microchip」と設定します。