

AKI-877ICE

Ver1.00.00

AKI-877ICEについて

AKI-877ICE は、MPALB-ICD と互換性のある、インサーキットエミュレーターです。

また、AKI-877ICE は書き込み機能も持っているため、PIC16F87 xシリーズを使用する場合はライターが不要です。

PC 側のソフトウェアは、Microchip Tchnology 社より供給されるMPLAB を使用します。

MPLAB のバージョンは、5.00.00以降が必要です。

MPLAB は Microchip Technology 社から供給されるPIC 統合開発環境の名称で、エディタ、アセンブラ、シミュレータ、エミュレータを統合したものです。AKI-877ICE は、MPLAB 上で、MPLAB-ICD Debugger(エミュレータとして動作します。

28pin タイプの PIC16F87x デバイスへ書き込み

AE-877ICE_L 基板に 28pin デバイスを挿入し、AE-877ICE_B 基板とDsub9 ピンコネクタどうしを接続し書き込みを行います。

AE-877ICE_L 基板の 40pin ヘッダーピンは、このあと説明するチュートリアルプログラム動作確認のために用意されています。

AE-877ICE とMPLAB-ICDの相違点

	AE-877ICE	MPLAB-ICD	備考
ICEへの動作電圧供給方法	AE-877ICE_HIに15Vを供給	ターゲットボードより3~5.5Vを供給	AE-877ICEで、電源ラインにダイオードブリッジを取り付けた場合は、16Vを供給。
マイコンのエミュレーション電圧範囲	5Vのみ	3~5.5V	
PIC16F87xの 28pinデバイスへの書き込み	AE-877ICE_L基板を使用することにより可能	MPLAB-ICD単独では不可	双方ともヘッダーボードを使用しない場合は可能。
クロックの供給	ヘッダー(AE-877ICE_HI)上の発振器およびターゲットボードから供給可能。	ターゲットボードからのみ供給可能。	

AKI-877ICEまたは、MPLAB-ICD を使用して、デバックを行う場合は、ソースコード作成、回路製作の際に、制約が発生します。ソースコード作成および回路製作するときは、注意が必要です。

1. スタックは、7段までしか使用できない。(通常は8段です。)
2. プログラムメモリ0x000 アドレスには、必ず nop コードを入れる。
3. RB6,7 端子は、インサーキットデバックおよび書き込みで使用するので、ユーザは使用できない。
4. MCLR 端子は、インサーキットデバックおよび書き込みで使用するので、共有できるようにする。
5. 動作周波数 32 KH から20 MH z
6. プログラムメモリ、レジスタにはユーザが使用できないエリアが存在します。

品種	使用不可レジスタ	使用不可プログラムメモリ
PIC16F870/871/872	0x70,0x0BB-0x0BF	0x06E0-0x07FF
PIC16F873/874	0x70,0x0EB-0x0F0	0x0EE0-0x0FFF
PIC16F876/877	0x70,0x1EB-0x1EF	0x1F00-0x1FFF

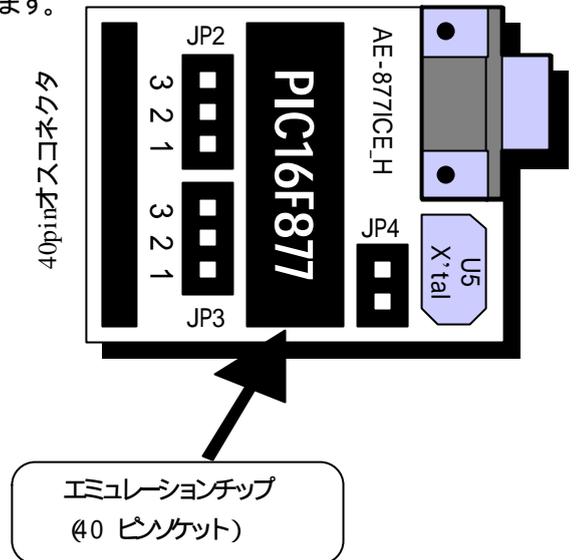
PICはMicrochip Technology Inc. の登録商標です。MPLAB、ICSPはMicrochip Technology Inc. の商標です。

AE-877ICE_H ヘッダーボード

ヘッダーボード(AE-877ICE_H)の PIC16F877 は、エミュレーションチップと呼ばれています。ユーザが作成したプログラムは、この PIC16F877 に書き込まれエミュレーションが行われます。デバック終了後は、このデバイスを取り外しターゲットボードに取り付けて完了です。(最終コードを書き込むときには Debug mode を disable で書き込みます。)
このエミュレーションチップには、AE-877ICE_B 基板より、+5V が供給されています。

JP4 ターゲットボードに、+5V を供給するときジャンパーを挿入します。
ターゲットボードとヘッダーボードで電源を分離するときはジャンパーをはずします。
今回は、ターゲットボード(AE-877ICE_L)に電源を供給するのでジャンパーを挿入します。

JP2,JP3 U5 には、水晶発振器を取り付けます。この発振器からエミュレーションチップにクロックを供給する時は、JP2 と JP3 の、2-3 にジャンパーを挿入します。
ターゲットボードからクロックを供給する時は、JP2 と JP3 の 1-2 にジャンパーを挿入します。
今回は、水晶発振器からクロックを供給するので 2-3 にジャンパーを挿入します。

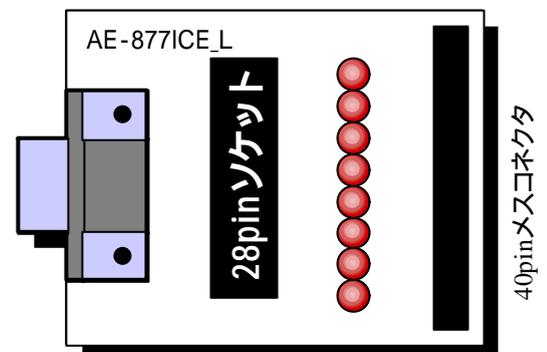


AE-877ICE_L デモボードについて

AE-877ICE_L デモボード基板は、2つの目的をもっています。
一つは、28ピンデバイスへのユーザが作成したコードのプログラミング、
もう一つは、チュートリアルで作成するプログラムの動作確認用の LED 表示です。

28ピンデバイスにコードをプログラミングするときは、
ベースボードとデモボードの Dsub9 ピン同士を接続し使用します。

チュートリアルのデモボードとして使用するとき、
ヘッダーボードの 40ピンオスコネクタと、デモボードの 40ピンメスコネクタを接続し使用します。



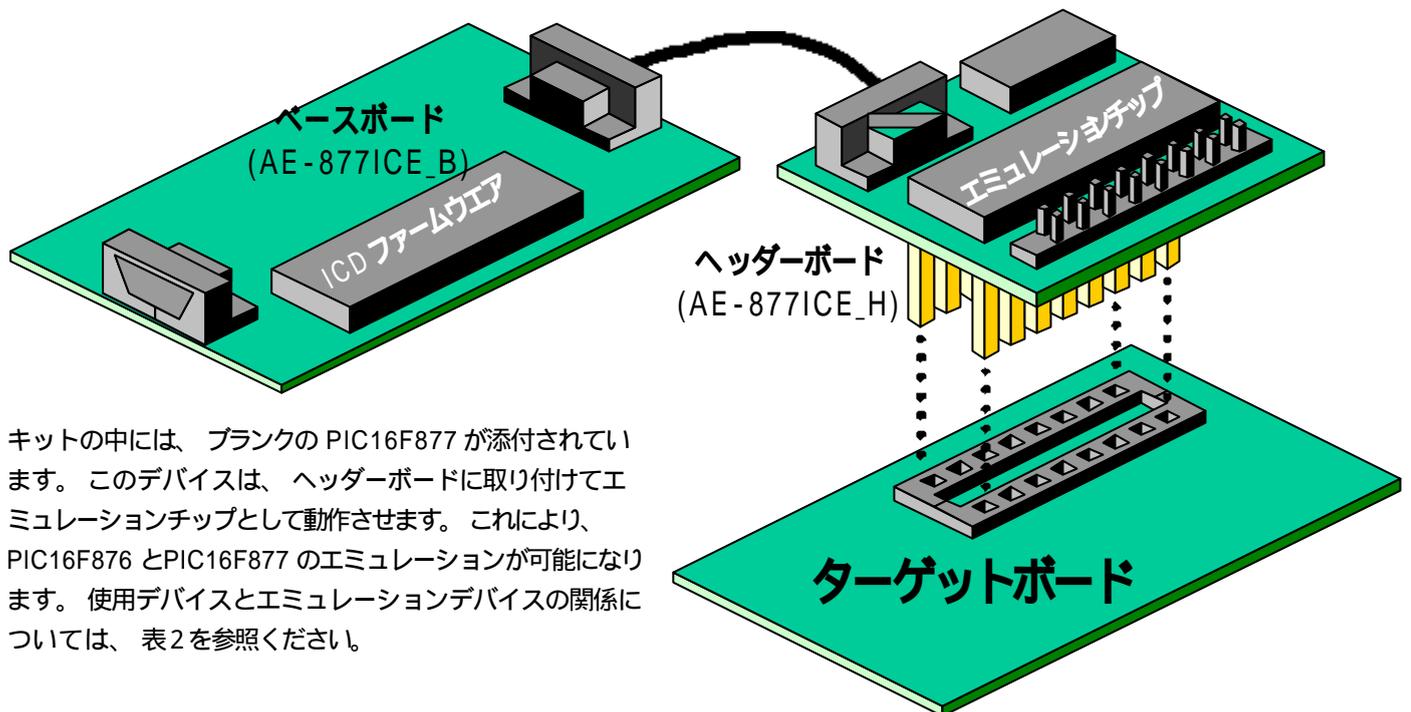
AKI-877ICEの基本的な使い方

AKI-877ICEの使い方は、2種類あります。

一つ目は、ヘッダーボード (AE-877ICE_H) を使用する方法で、もう一つは、ヘッダーボードを使用せずに、ユーザのターゲットボードを直接使用する方法です。

ヘッダーボードを使用すると、ターゲットボードに手を加える必要が無いので手軽ですが、エミュレーションできるデバイスに制限があります。

ヘッダーボードを利用する方法



キットの中には、ブランクのPIC16F877が添付されています。このデバイスは、ヘッダーボードに取り付けてエミュレーションチップとして動作させます。これにより、PIC16F876とPIC16F877のエミュレーションが可能になります。使用デバイスとエミュレーションデバイスの関係については、表2を参照ください。

表2 ヘッダーボード上のデバイスと、エミュレーション可能デバイスの関係

使用するデバイス		ヘッダーボードに取り付けるデバイス	デバイスへのプログラミング
PIC16F870	28ピン/FLASH 2Kワード	PIC16F871	デモボード (AE-877ICE_L) の Dsub9コネクタ側を使用。
PIC16F871	40ピン/FLASH 2Kワード	PIC16F871	ヘッダーボード(AE-877ICE_H)を使用。
PIC16F873	28ピン/FLASH 4Kワード	PIC16F874	デモボード (AE-877ICE_L) の Dsub9コネクタ側を使用。
PIC16F874	40ピン/FLASH 4Kワード	PIC16F874	ヘッダーボード(AE-877ICE_H)を使用。
PIC16F876	28ピン/FLASH 8Kワード	PIC16F877	デモボード (AE-877ICE_L) の Dsub9コネクタ側を使用。
PIC16F877	40ピン/FLASH 8Kワード	PIC16F877	ヘッダーボード(AE-877ICE_H)を使用。

2000.06.30時点ではPIC16F870/871はリリースされておりません。
したが、AE-877ICEは、PIC16F870/871をサポートする予定ですが動作確認は行われておりません。

ターゲットボードを利用する方法

以下に、ヘッダーボード (AE-877ICE_H) を使用せずに、直接ユーザが製作したターゲットボードにベースボード (AE-877ICE_B) を接続するための参考図を記します。この場合、ターゲットボード上の PIC16F87x がエミュレーションチップとして動作します。全 PIC16F87x シリーズのデバイスが使用可能です。

Dsub 9 ピンメスコネクタの 6pin と 7pin について

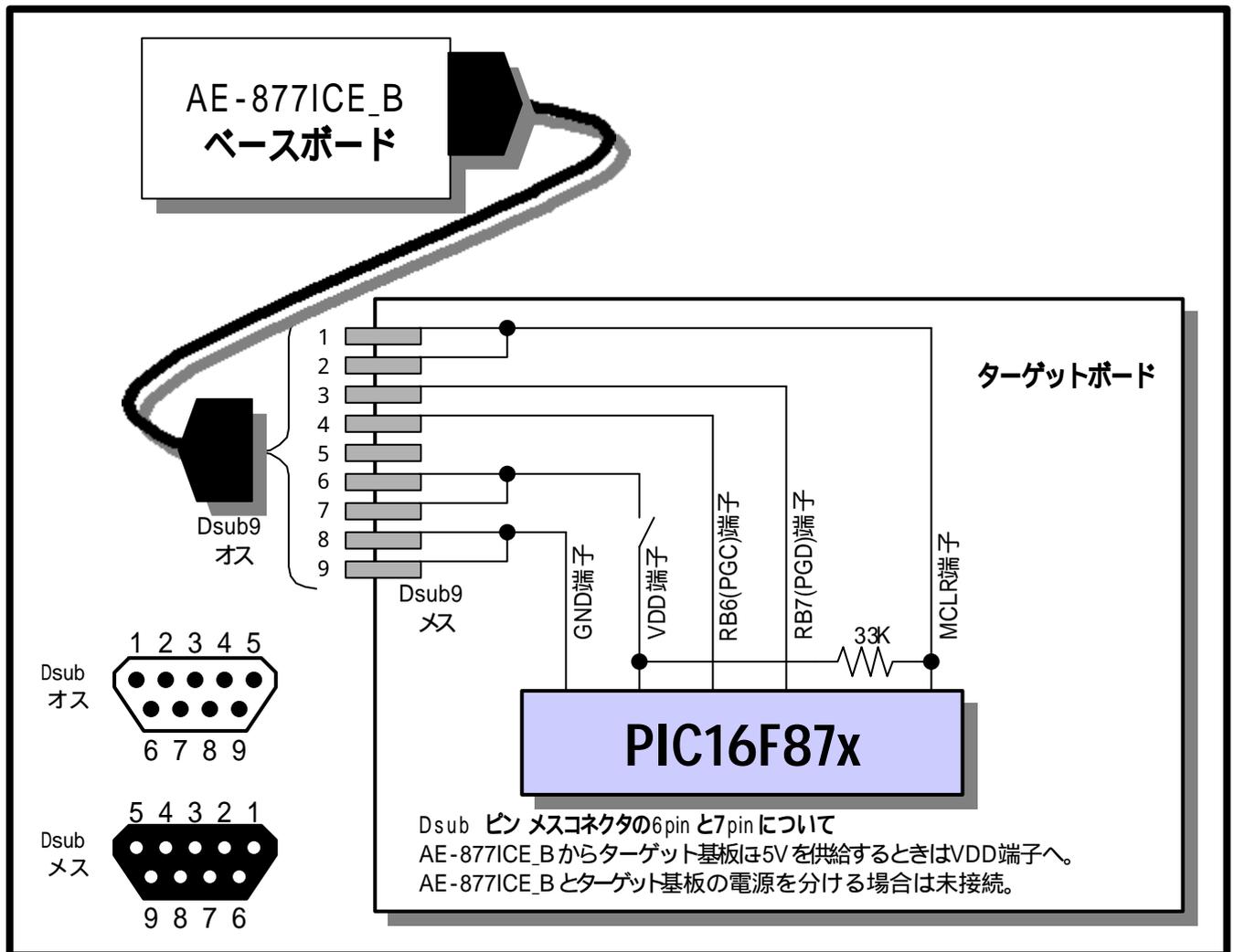
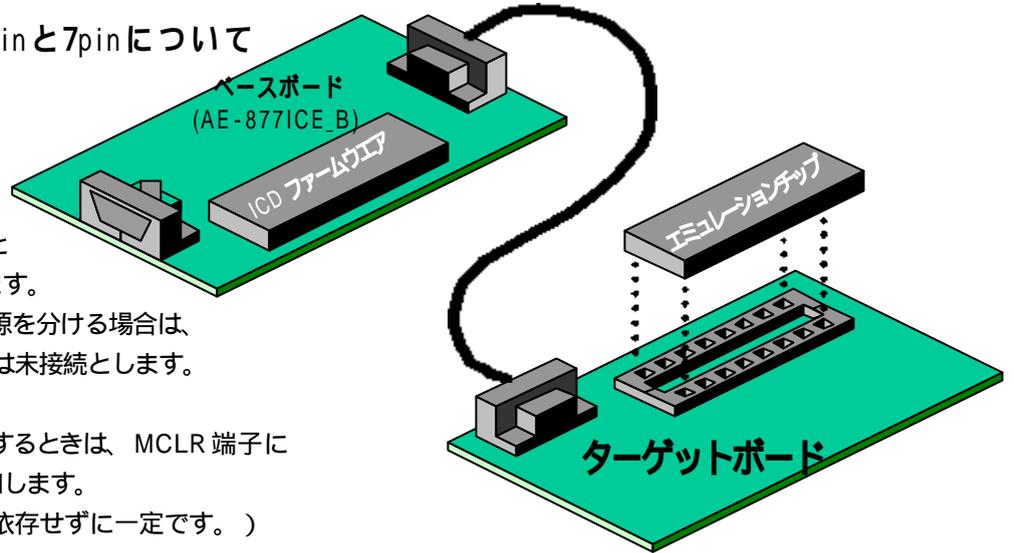
AE-877ICE からターゲット基板に +5V を供給するときは、

Dsub 9 ピンメスコネクタの 6, 7pin と PIC16F87x の VDD 端子を接続します。

AE-877ICE とターゲット基板の電源を分ける場合は、Dsub 9 ピンメスコネクタの 6, 7pin は未接続とします。

(デバイスにコードをプログラミングするときは、MCLR 端子に VPP 電圧 (12~13.5V) を印加します。

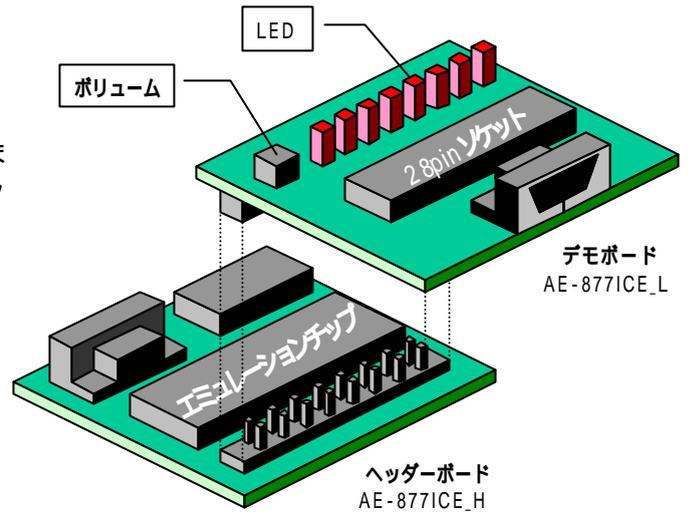
この VPP 電圧は、VDD 電圧には依存せず一定です。)



チュートリアル

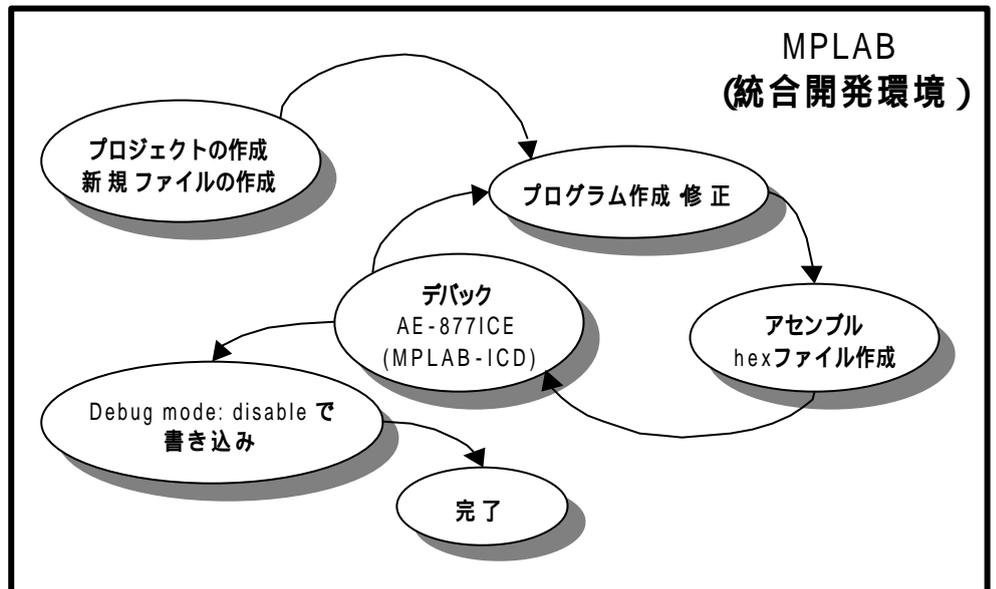
ここでは、PIC16F877 を使用して、RA0/AN0 でAD変換を行い、結果をPORTC に出力するプログラムを作成します。デモボード上のボリュームを回すことでLEDの表示データが変わります。ヘッダーボードと、デモボードを右図のように40ピンコネクタで接続します。

ここではデモボードの、Dsub9 コネクタ、28ピンソケットは使用しません。



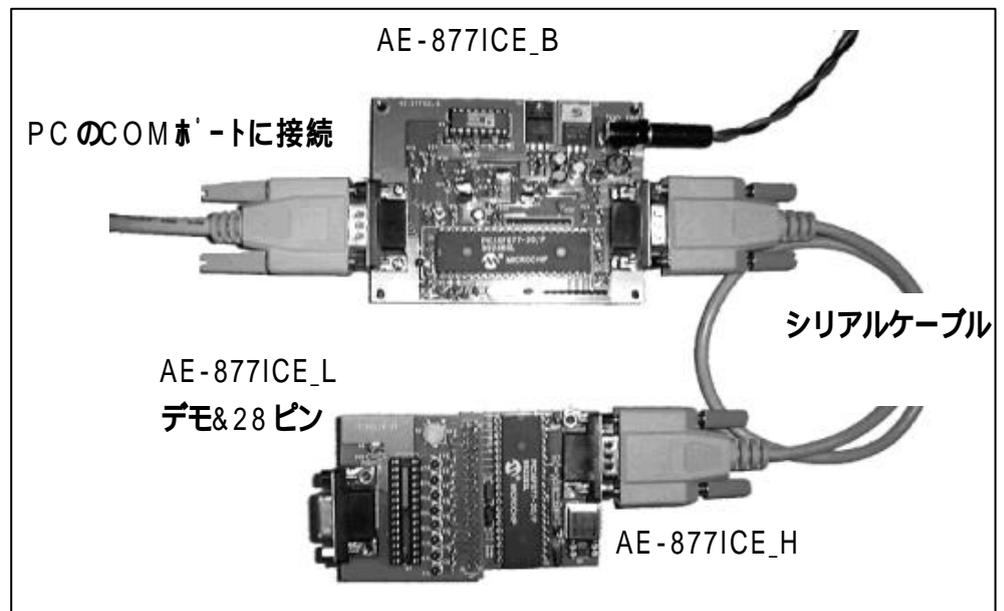
デバック作業の流れ

MPLAB を使用したデバック作業の流れを右図に記します。ソースコードの作成からデバイスへのプログラミングまでMPLAB 内で可能です。



使用方法

1. MPLAB をインストールします。
MPLAB は、
C:¥Program Files¥Mplab にインストールされます。
2. C ドライブの下に、
フォルダー Tutorial を作成します。
(C:¥tutorial)
3. AKI-877ICE をPC に接続します。



4. MPLAB を起動します。

メニューバーの項目の全てが選択可能であることを確認してください。もし、非選択の項目がある場合はMPLAB を再起動させてください。（プルダウンメニューの中に非選択項目があっても問題ありません。）

MPLAB は、プロジェクトの考えを採用しています。このプロジェクトが、ソースファイルや、エミュレーションに必要なデバッグ情報ファイルを管理します。

これより、新しくプロジェクトを作成する方法を説明します。

最初に、メニューバーの[File]>[New]を選択します。<Untitled xウィンドウ> が開きます。（最終的に、このウィンドウにPIC のアセンブラソースコードを記述します。）



"A Project is not currently opened. Would you like to create a new project?" のメッセージが表示されます。

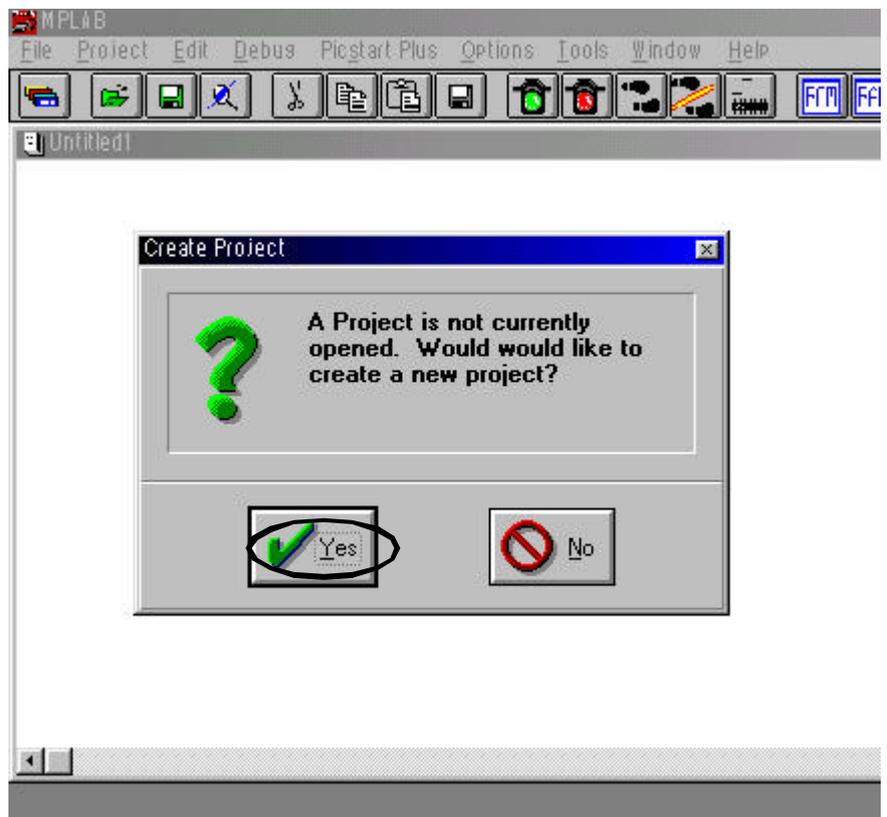
ここでは、新しいプロジェクトを作成しますので [Yes] を選択します。

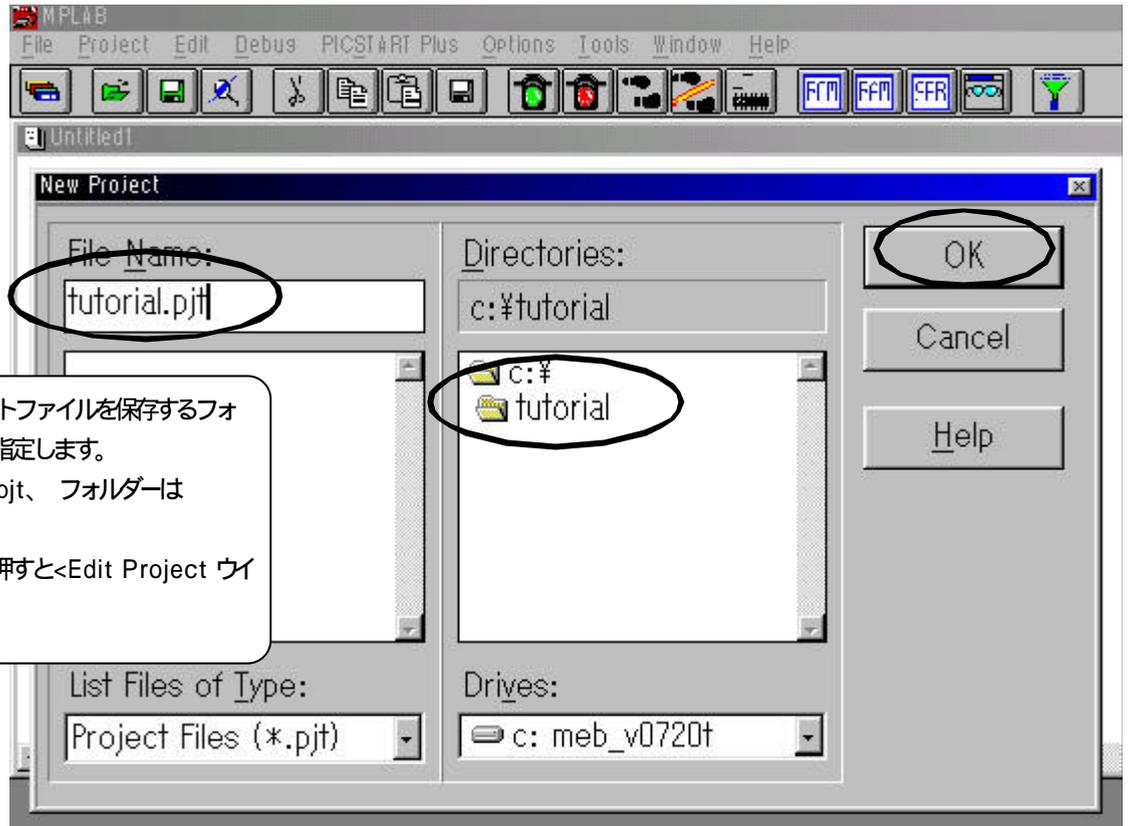
プロジェクトファイルを保存する画面が現れますので、ファイル名はtutorial.pjt、保存フォルダーは、C:\Tutorial として保存します。

次に、Edit Project ウィンドウが現れます。ここでは、プロジェクトに関する各種設定を行います。

設定する内容は、以下のとおりです。

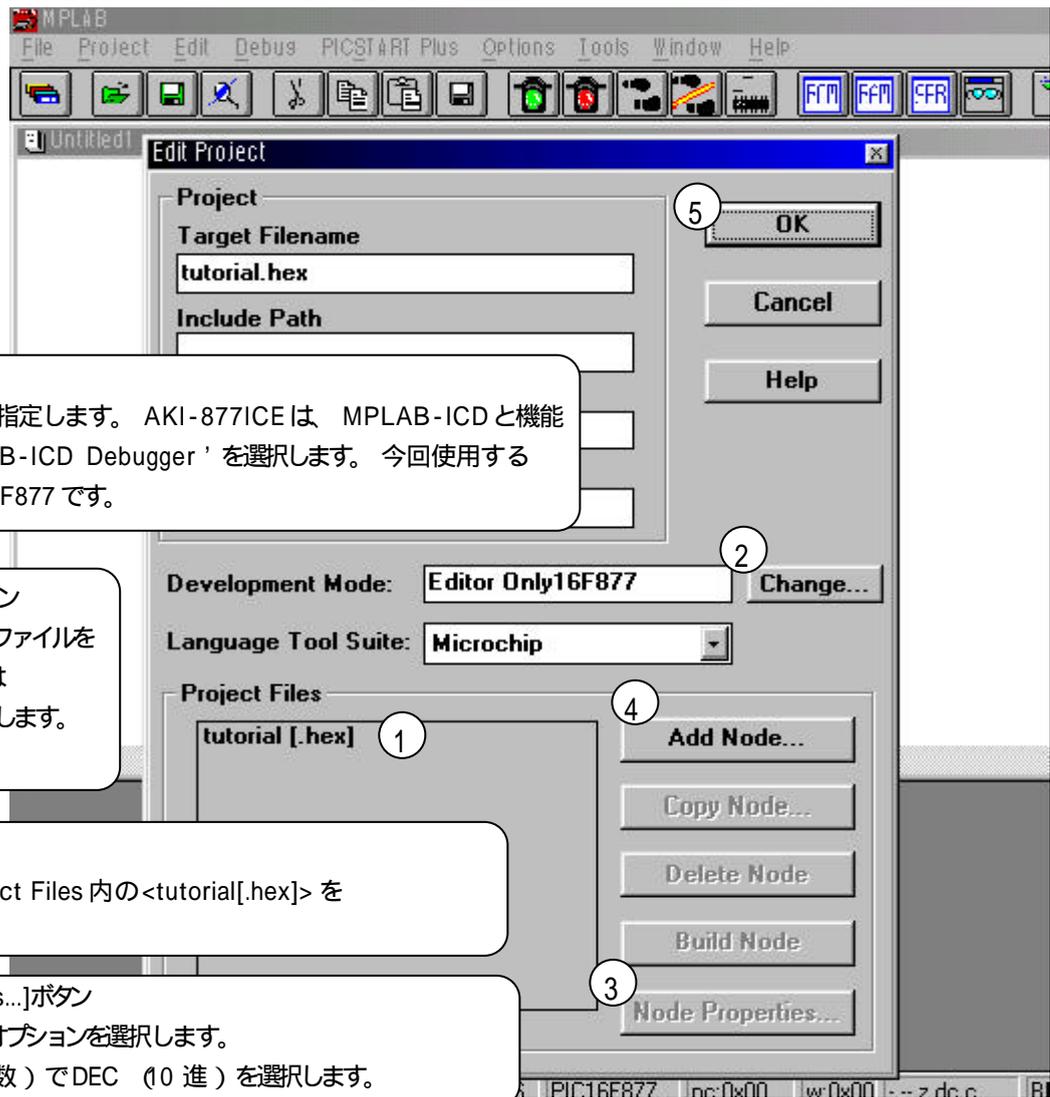
1. 使用する開発環境
2. アセンブル時の設定
3. ソースファイルの設定





ここでは プロジェクトファイルを保存するフォルダーとファイル名を指定します。
 ファイル名はtutorial.pjt、フォルダーはtutorialです。
 最後に[OK]ボタンを押すと<Edit Project ウィンドウ> が現れます。

番号が振られている順番で作業を行います。



[Change...]ボタン
 使用する開発環境を指定します。AKI-877ICEは、MPLAB-ICDと機能互換なので 'MPLAB-ICD Debugger' を選択します。今回使用する Processor はPIC16F877 です。

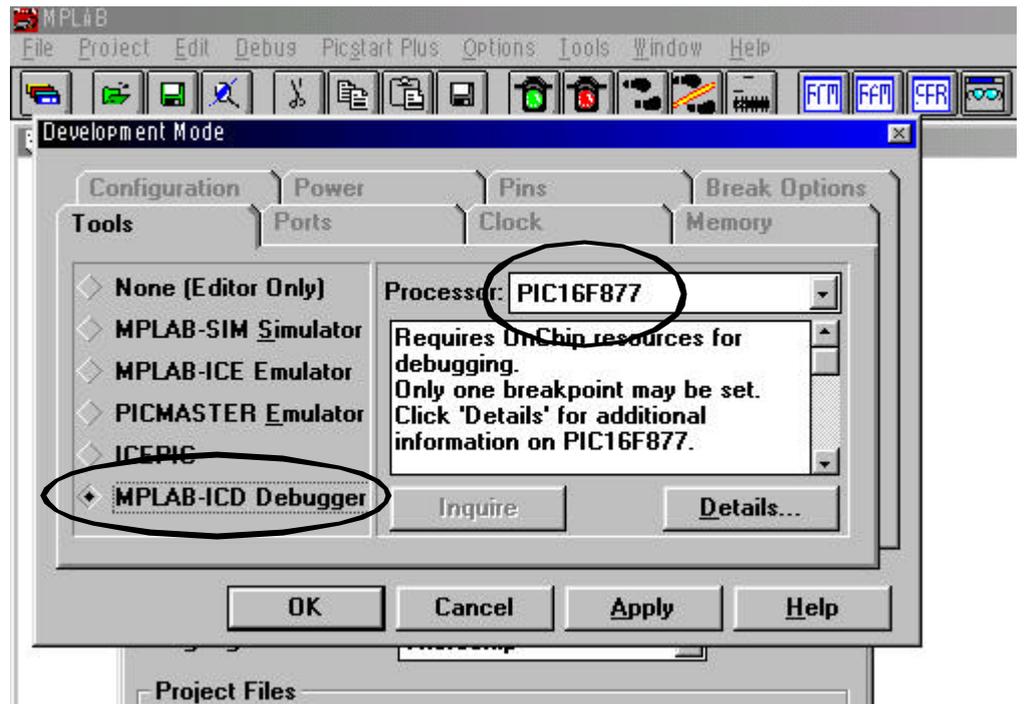
[Add Node...]ボタン
 アセンブルするソースファイルを指定します。今回は tutorial.asm を指定します。

重要
 最初に必ず、Project Files 内の<tutorial[.hex]> を選択しておきます。

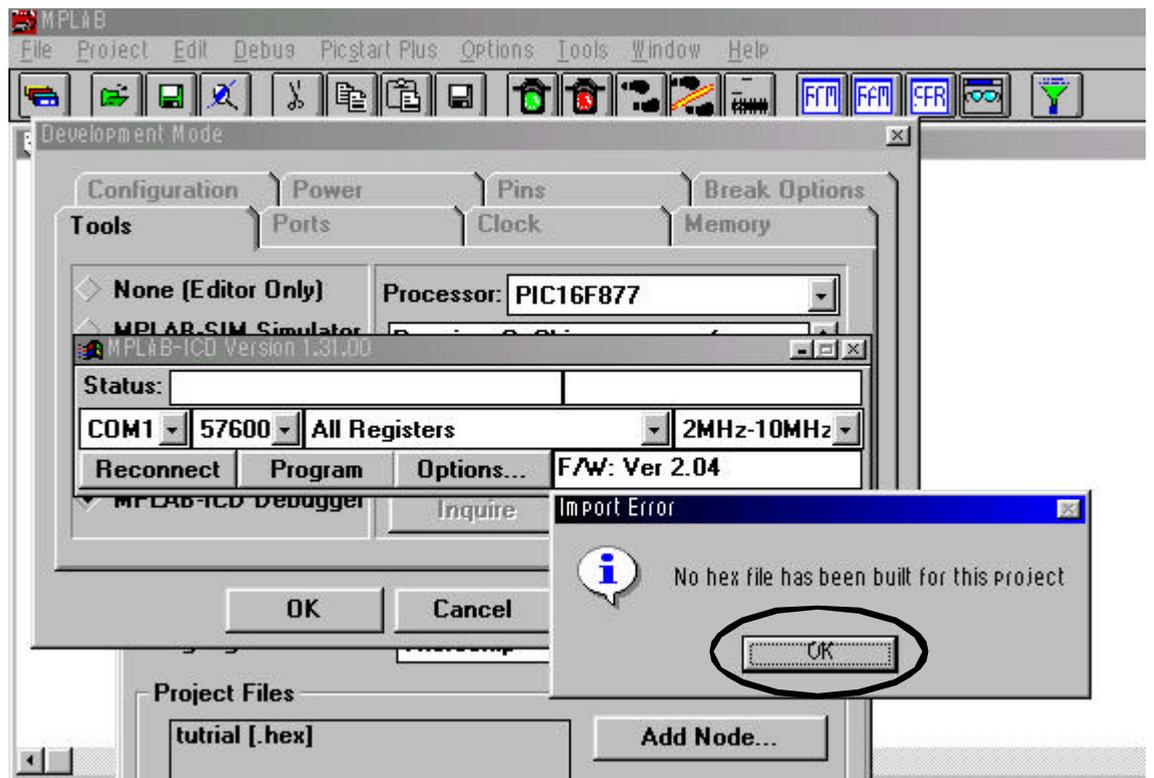
[Node Properties...]ボタン
 ここではアセンブラのオプションを選択します。
 Default Radix (基数) でDEC (10進) を選択します。

前ページの<EditProject画面>で [Change...] ボタンを押したときに現れる<DevelopmentMode>画面

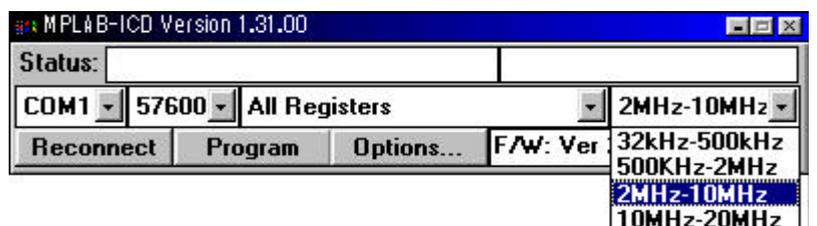
ここでは 877_ICE を使用するの
で、互換品であるMPLAB-ICD
Debugger を選択します。
また、Processor は、PIC16F877
を選択します。



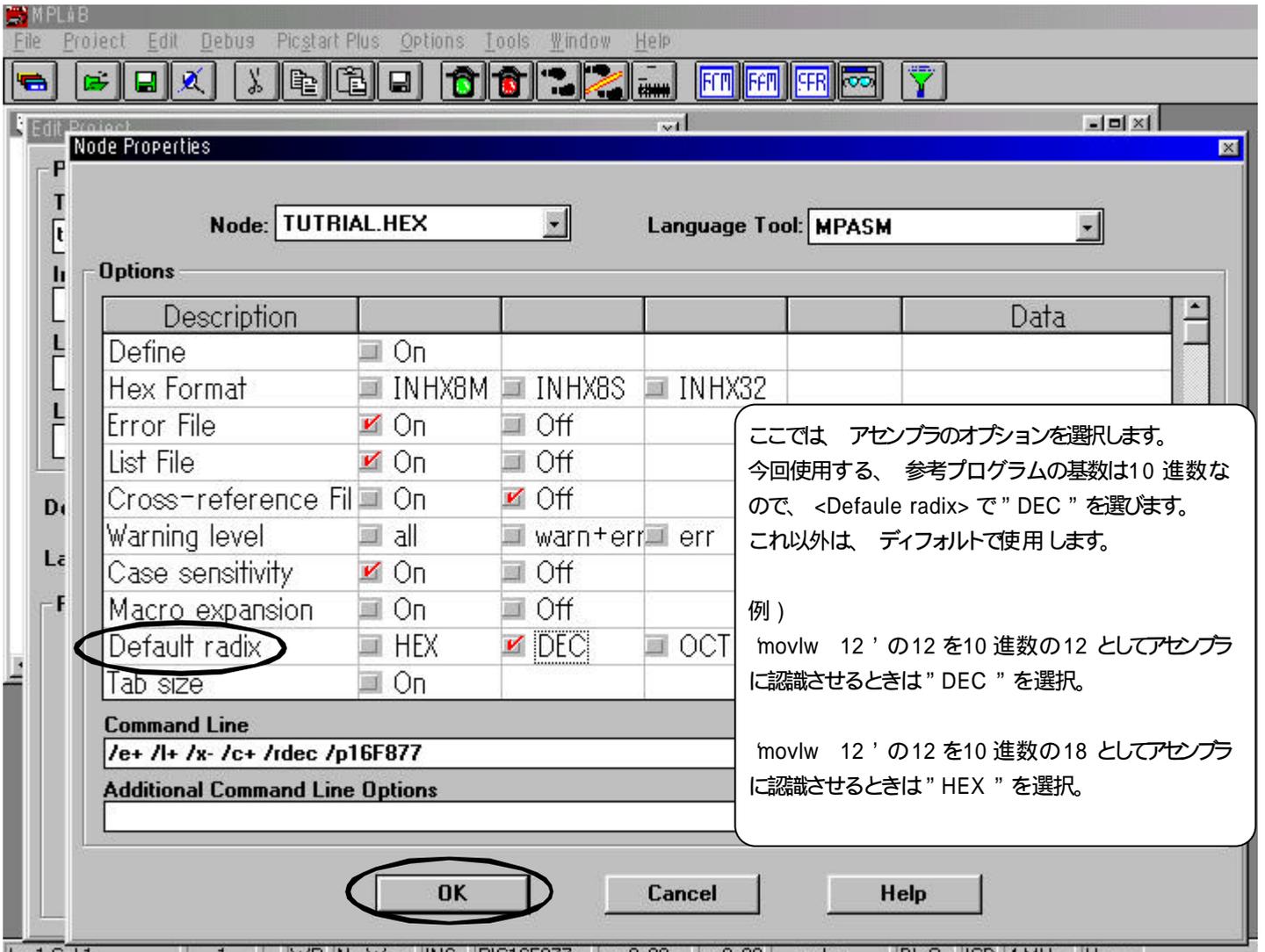
上図の<Develop-
ment Mode 画面>で
[OK]ボタンを押したと
き この時点では
オブジェクトファイル
*.hex が生成されてい
ないので、下図の
<MPLAB-ICD
Version x.xx.xx ウィ
ンドウ> と同時に
<Import Error ウィン
ドウ> ができます。
[OK]ボタンを押して次
に進んでください。



右ウィンドウでは、通信ポート、通信スピー
ド、デバック時に呼び出すレジスタの種類(少
ない方がMPLABの応答が良くなる)ターゲッ
トデバイスに供給するクロックの周波数を設
定することができます。
ターゲットデバイスに16MHzを供給する
ときは、"10MHz-20MHz" を選択します。



前々ページの<Edit Project 画面>で [NodeProperties...] ボタンを押したときに現れる <NodeProperties>画面



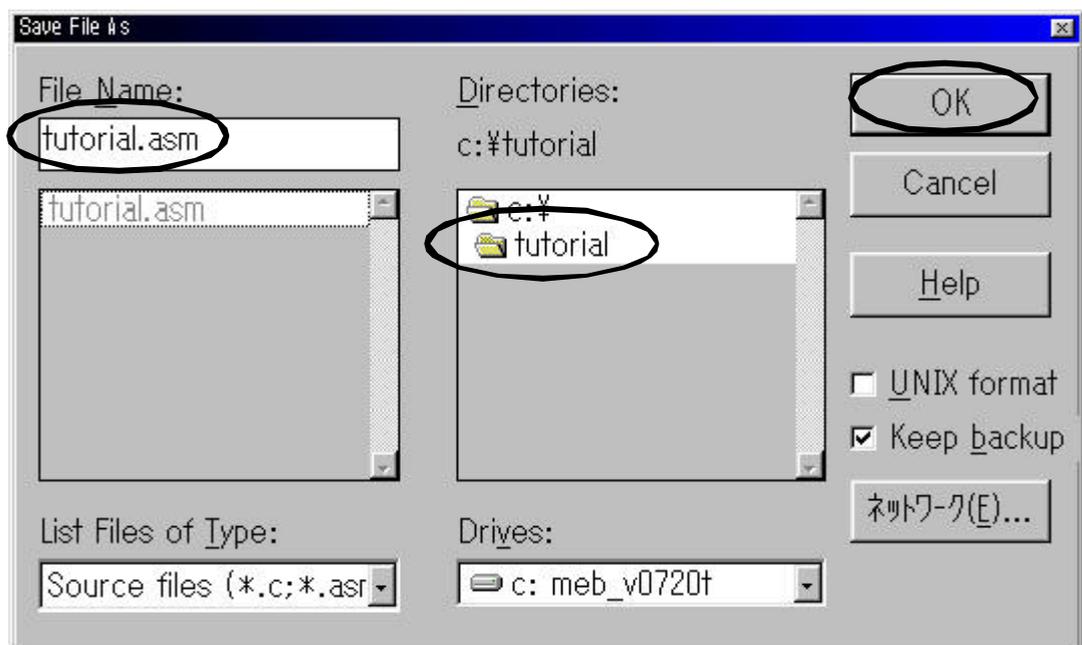
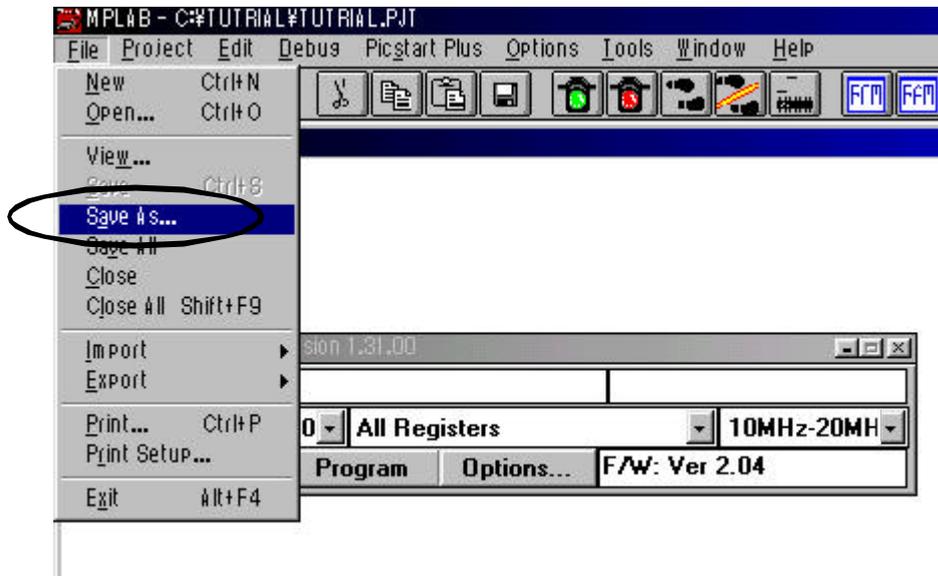
前々ページの<Edit Project 画面>で [AddNode...] ボタンを押したときに現れる<Add Node>画面



ソースファイルの作成

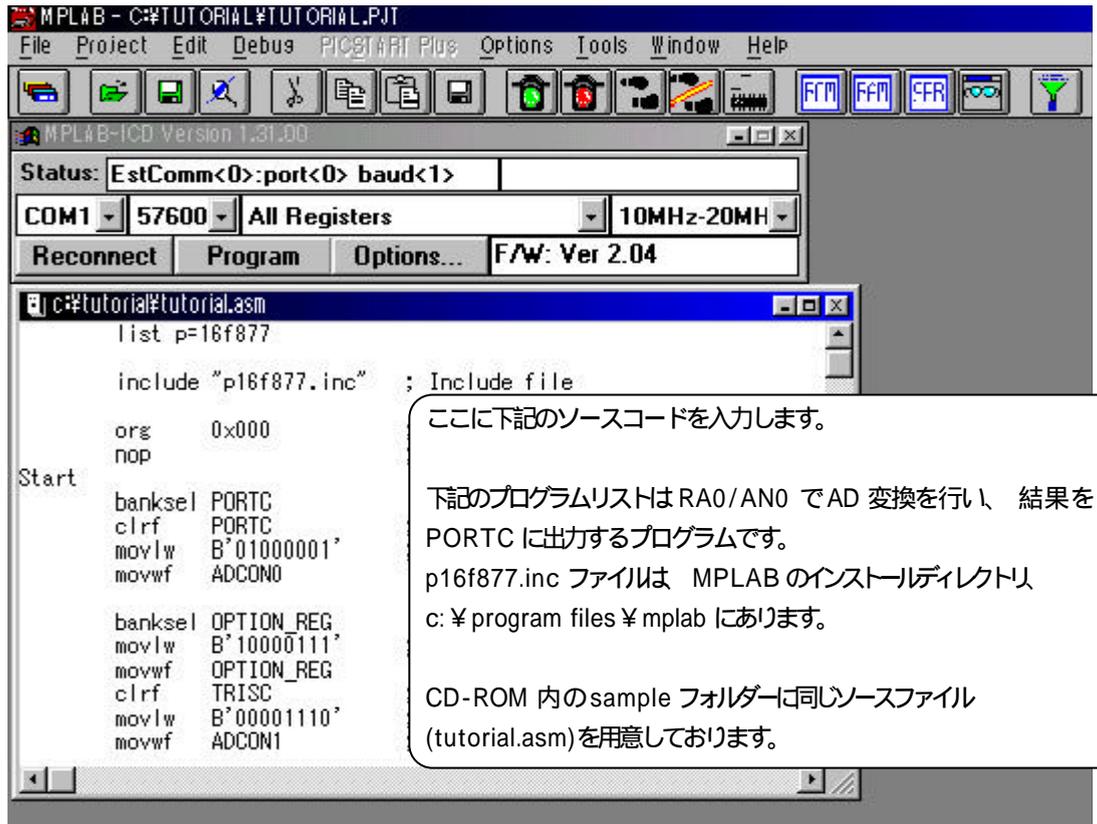
<Untitled xウィンドウ > に tutorial.asm という名前をつけて保存します。

ここで、初めてソースファイルがハードディスク上に作られます。



ソースコードの入力

ソースコードを入力します。



```

list p=16f877

include "p16f877.inc"           ; Include file

org      0x000                 ; Start at the reset vector
nop                                           ; limitation for IDC

Start

banksel  PORTC
clrf     PORTC                 ;Clear PORTC
movlw   B'01000001'          ;Fosc/8, A/D enabled
movwf   ADCON0

banksel  OPTION_REG
movlw   B'10000111'          ;TMR0 prescaler, 1:256
movwf   OPTION_REG
clrf    TRISC                 ;PORTC all outputs
movlw   B'00001110'          ;Left justify,1 analog channel
movwf   ADCON1               ;VDD and VSS references

Main

banksel  PORTC

btfss   INTCON,TOIF           ;Wait for Timer0 to timeout
goto    Main
bcf     INTCON,TOIF

Wait

bsf     ADCON0,GO             ;Start A/D conversion
btfss   PIR1,ADIF            ;Wait for conversion to complete
goto    Wait

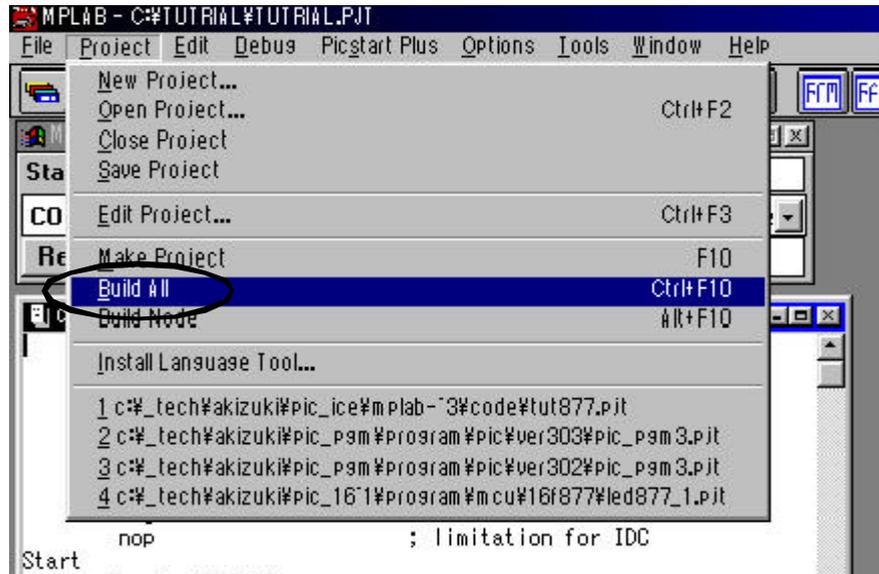
movf    ADRESH,W              ;Write A/D result to PORTC
movwf   PORTC                ;LEDs

goto    Main                  ;Do it again

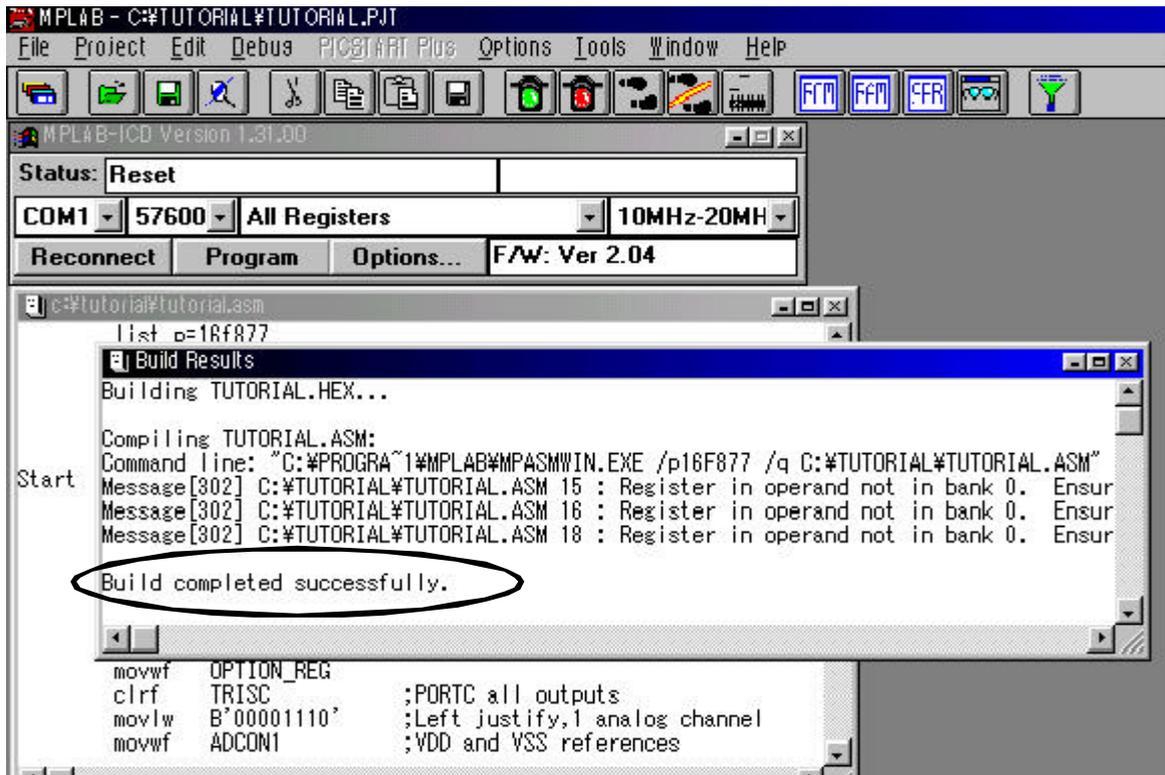
end
    
```

ソースファイルのアセンブル (ビルド作業)

前ページで入力したソースコードをアセンブルし、 オブジェクトファイル” tutorial.hex ” を生成するにはビルドを行います。



ビルドが終了すると、 <Build Results ウィンドウ> が現れ、 ビルド結果を表示します。



最終行に " Build completed successfully " が記述されていますので、 正常にオブジェクトファイル” tutorial.hex ” が生成されたことがわかります。

Message[302]がありますが、 これは注意を促すものです。[302]の場合、 異なるバンクのレジスタにアクセスしているので、 確実にバンク切り替えを行って下さい。」 というメッセージです。

エミュレーションの実行

エミュレーションを行うためには、生成された、tutorial.hex をエミュレーションチップに書き込む必要があります。 <MPLAB-ICD Version x.xx.xx ウィンドウ> の [Program] ボタンを押します。 これによりエミュレーションチップに tutorial.hex コードが書き込まれます。 この作業は アセンブル (ビルド) ごとに行います。



メニューボタンの変更

メニューボタンの種類は複数用意されています。 自分の使いやすいものを選びます。 ここでは "メニューボタンB" を選んでください。



Run ボタンを押すとプログラムが走ります。

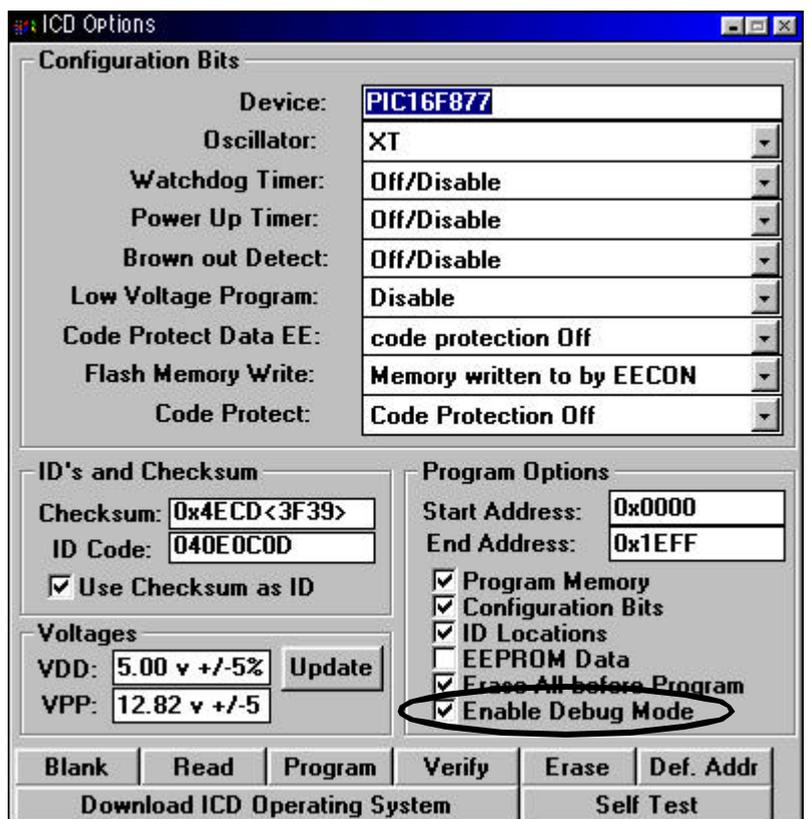
詳しい、MPLAB の使い方については、Microchip Technology 社より提供されているマニュアルをご参照ください。

<MPLAB-ICD Version x.xx.xx ウィンドウ > の Options... ボタンを押すと <IDE Options ウィンドウ > が現れます。

ここでは、書き込みのオプションを設定することができます。

重要

デバックが終了し、最終コードの書き込み時は、Program Options の Enable Debug Mode のチェックをはずしてください。



書き込み電圧の確認

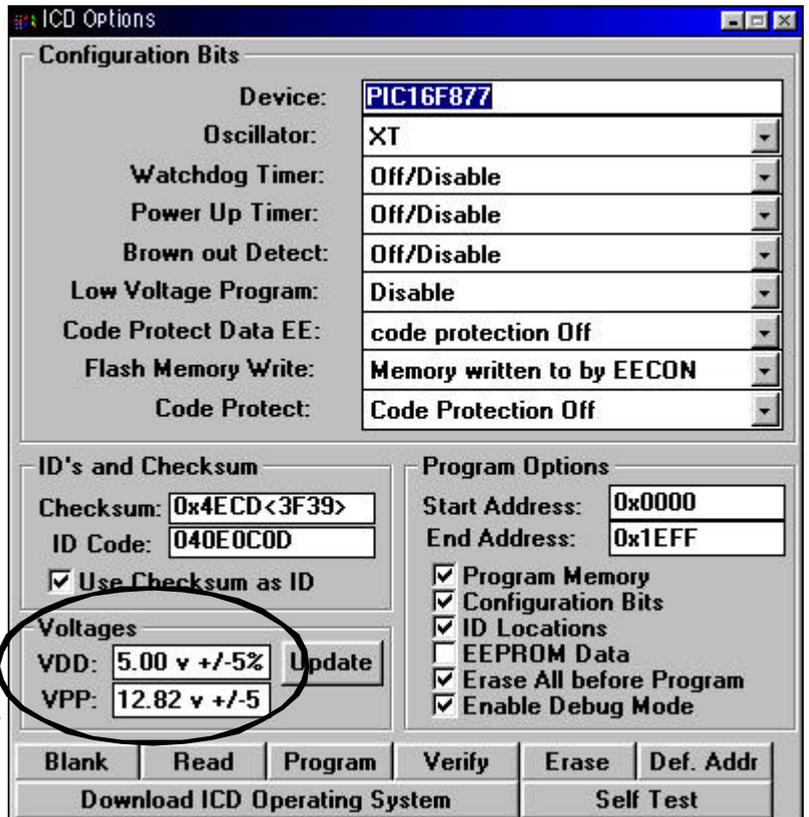
<Configuration Bits ウィンドウ>の Voltages で、VPP 電圧が、12.0V ~ 13.5V の範囲内に収まっていない場合。

ベースボード (AE-877ICE_B) の、U4 LM317 の 2 ピンと GND 間の電圧を測定し、12.0V ~ 13.5V の範囲内に収まっていない場合は、R33(120)、R34 (1.1K)を確認してください。

実測電圧が、12.0V ~ 13.5V の範囲内に収まっている場合は、問題ないので<Configuration Bits ウィンドウ>の Voltages の VPP 電圧値は無視してください。

備考

エミュレーションチップを動作させる VDD と VPP は、MPLAB-ICD の場合、ターゲットボードが 3 ~ 5.5V の電圧を与え、VDD はターゲットボードに供給される電圧をそのまま使用し、VPP はこの電圧を昇圧して VPP=12.0 ~ 13.5V を作り出してエミュレーションチップに供給します。この場合、ターゲットボードから供給される電圧が、何ボルトであるかが分からないと VPP 電圧を作り出すことができません。そこで発光ダイオードの電圧降下を利用し、これを基準電圧として、<Configuration Bits ウィンドウ>の Voltages の VPP/VDD 電圧値を求めています。



AKI-877ICE の場合、15V の電圧をベースボードに与え、レギュレータ(LM7805、LM317)によって VDD=5V、VPP=12.0 ~ 13.5V を作り出しエミュレーションチップに供給します。AE-877ICE は、MPLAB-ICD と同じ方法で VDD、VPP 電圧を読み取っていますが、VPP 電圧は、レギュレータで作りますのでこの電圧にズレが生じても問題ありません。発光ダイオードの電圧降下の特性は、種類、ロット、温度により変化しますので、実際の電圧値と表示に大きな差が生じる場合があります。