

MPLABの使い方

2003年 11月



丸紅ソリューション株式会社

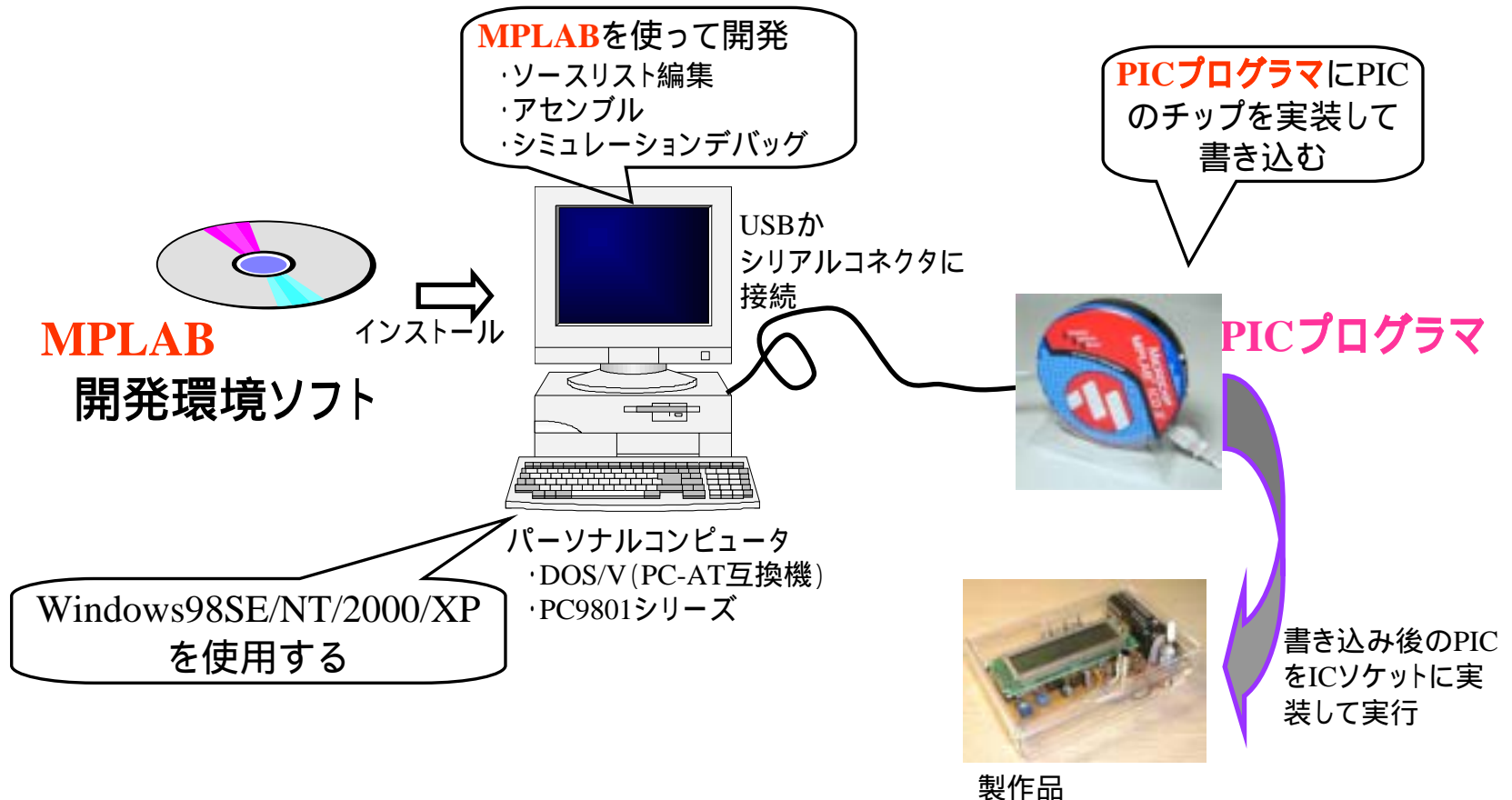
MPLABの使い方

- ◆MPLABの概要
- ◆MPLABを起動
- ◆プロジェクトの新規作成
- ◆ソースファイルの作成と登録
- ◆アセンブル実行

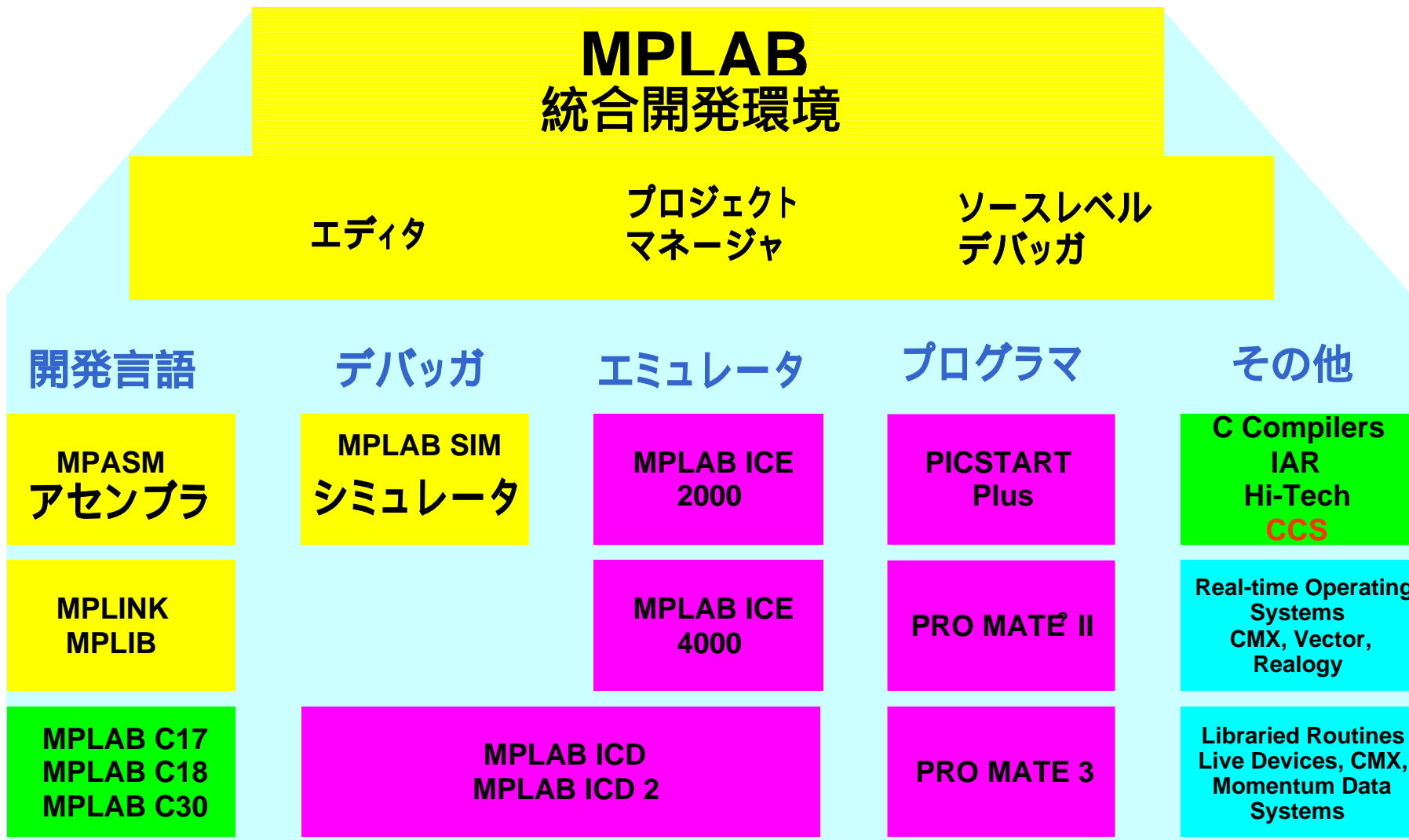
PIC開発環境の構築

◆最小の開発環境

- ◆プログラム開発は全てパソコン上で行う
- ◆PICプログラマでプログラムを書き込む



統合開発環境 MPLABの体系



MPLABの特徴

- ◆ プログラム開発の全ステップのツールを包含
 - ◆ プログラムソース入力エディット
 - ◆ アセンブル
 - ◆ シミュレーションデバッグ
 - ◆ PICプログラマサポート
 - ◆ ハードウェアシミュレータ / エミュレータサポート
- ◆ フリーソフトで提供されている
- ◆ 全てのPICのシリーズに対応
 - ◆ 純正プログラマのバージョンアップが可能
(MPLABに添付されている)
- ◆ 他の言語コンパイラも統合できる
 - ◆ HiTech-C **CCS-C** ...

開発環境のバージョンアップ

◆ MPLABのメジャーバージョンアップ

◆ Ver 5.70 **Ver 6.10** '02/11月

Ver 6.12 '03/2月

Ver 6.20 '03/4月

Ver 6.30 '03/7月

Ver 6.32 '03/9月

Ver 6.XX

- ◆ 16ビットから32ビットアプリケーションへ
- ◆ デバッグ機能強化(ソースコードデバッグ)
- ◆ C言語とアセンブラの混在デバッグ
- ◆ dsPIC対応
- ◆ USB接続のICD2サポート

MPLAB®のバージョン差異

| | MPLAB 5.70 | MPLAB 6.10 | MPLAB 6.32 |
|----------------|------------|------------|------------|
| dsPIC | No | Yes | Yes |
| PIC18 | Most | Yes | Yes |
| PIC17 | Yes | No | Yes |
| PIC16 midrange | Yes | Yes | Yes |
| PIC16 baseline | Yes | No | Yes |
| PIC14 | Yes | No | No |
| PIC12 | Yes | No | Yes |
| rfPIC | No | Yes | Yes |

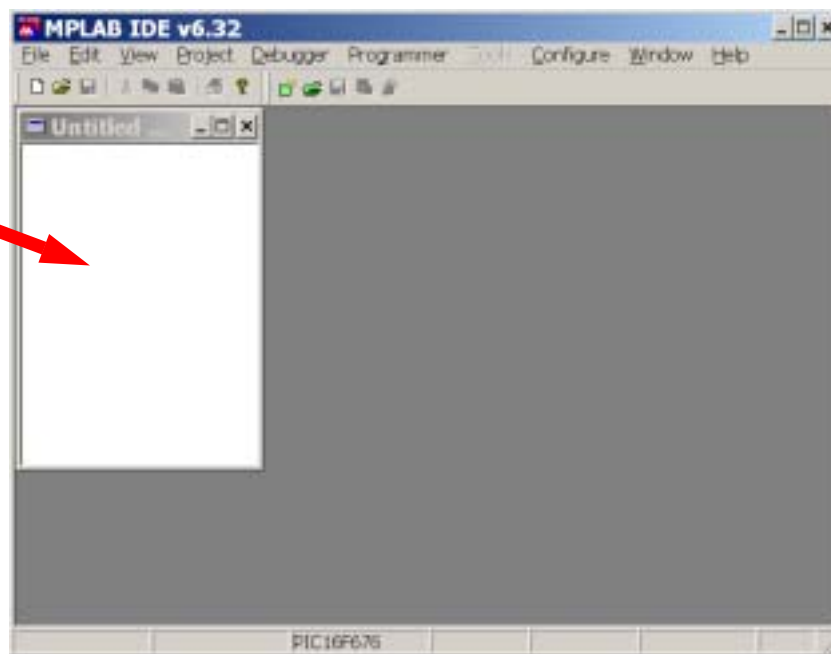
プロジェクトの作成

- ◆MPLABを起動
- ◆Project Wizardで作成

MPLABの起動

- ◆ スタート・ボタンを押して、スタート・メニューを表示させます。
- ◆ プログラム Microchip MPLAB IDE MPLABを起動します。
(またはMPLABのアイコンをダブルクリック)

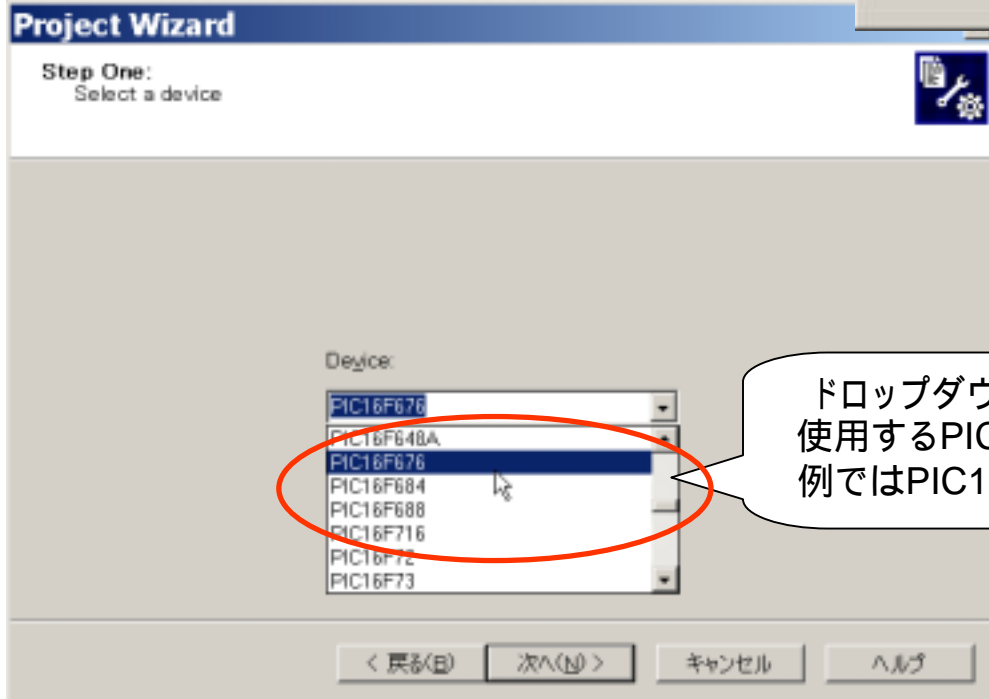
Project Windowと呼ばれ
プロジェクトに関連する
ファイルの一覧を表示



プロジェクトの作成(1)

◆ Project Project Wizard Welcom画面 PICの種別指定

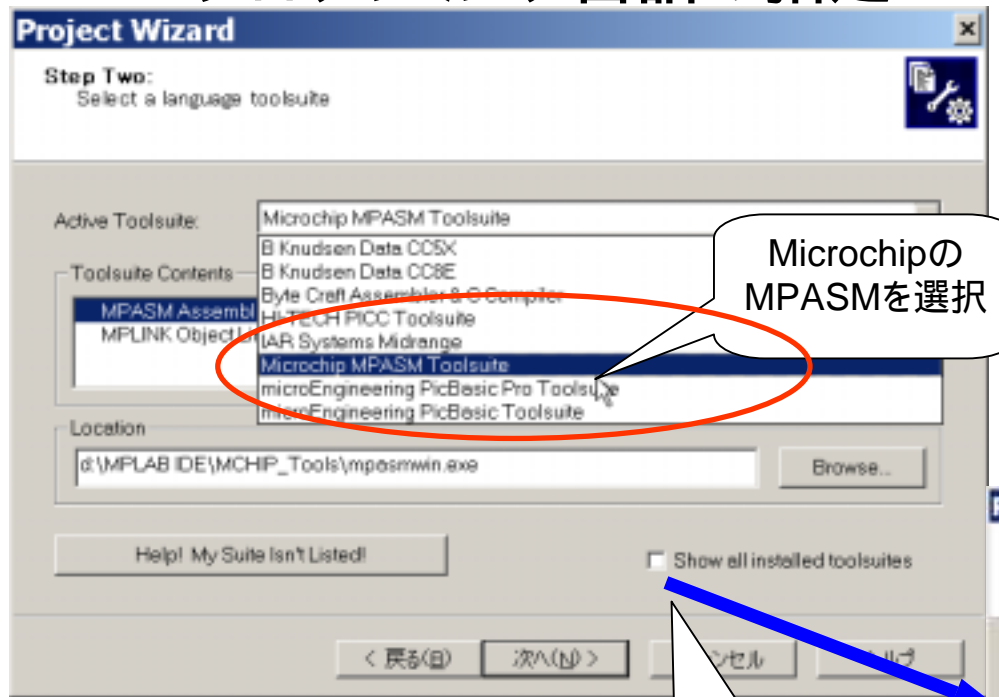
PICデバイス種別指定



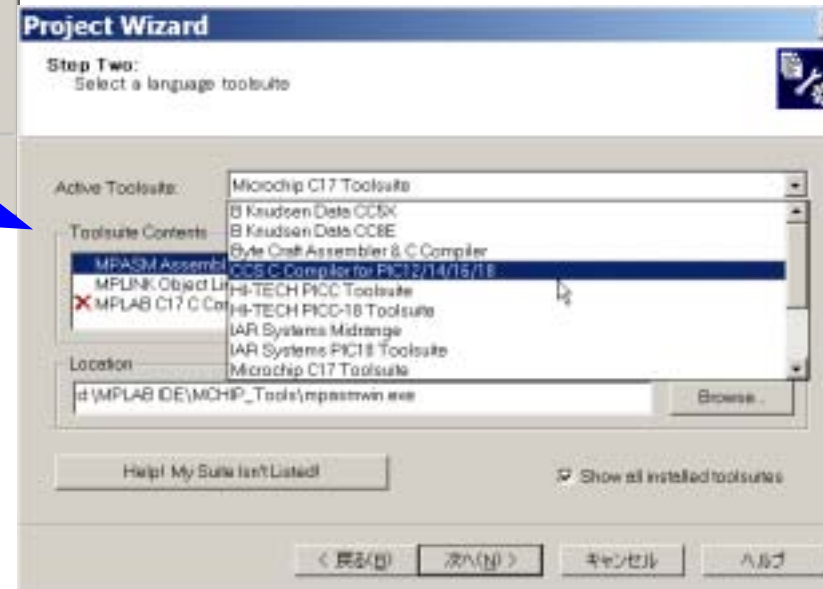
ドロップダウンリストから
使用するPICを選択する。
例ではPIC16F676を選択

プロジェクトの作成(2)

プログラミング言語の指定



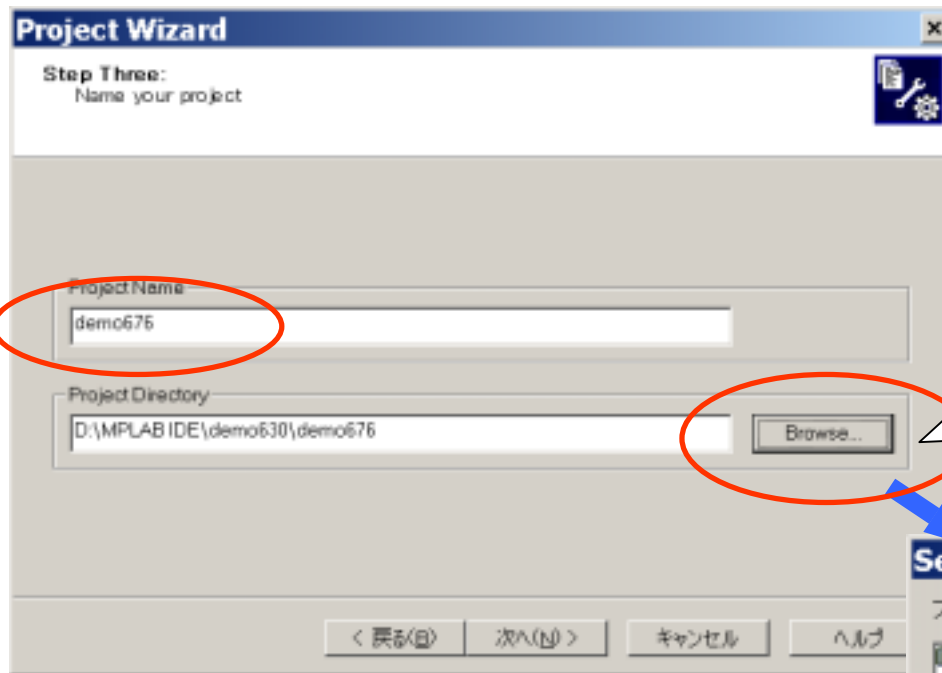
ここにチェックを入れると、他社の言語もドロップダウンリストに表示されるようになる



プロジェクトの作成(3)

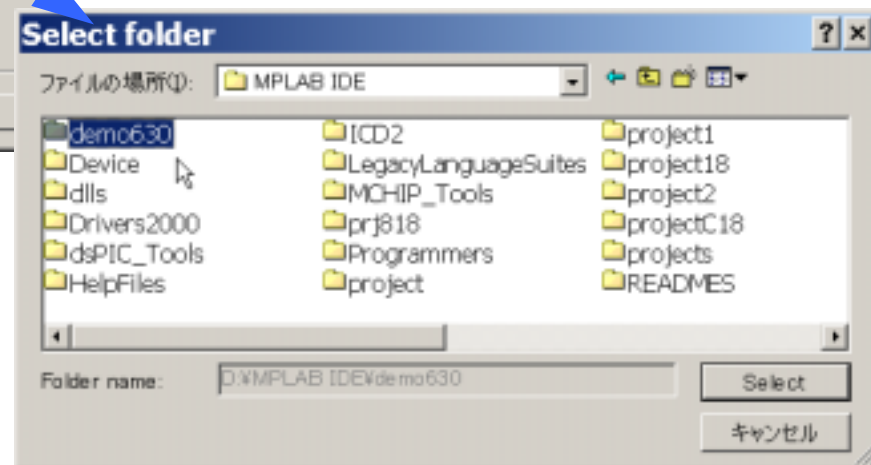
プロジェクト名とフォルダ指定

- ◆ 指定フォルダが無い時は自動的に生成する



Browseボタンで
フォルダ設定画面が開く
のでそこで選択。

新規フォルダの時は
直接文字入力



プロジェクトの作成(4)

ソースファイルの登録

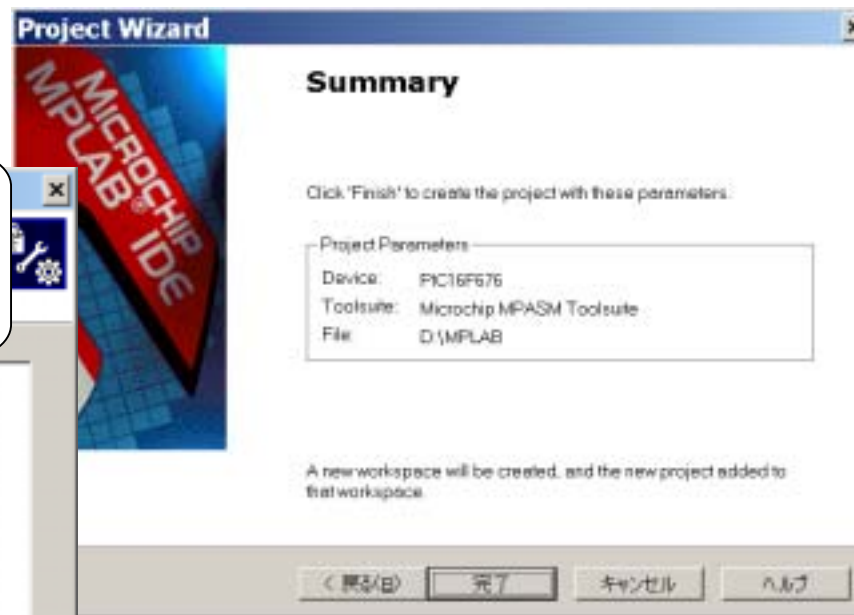
新規作成の場合には何もせず「次へ」

既にソースファイルがある場合には「Add」ボタン

確認画面

設定内容の確認

ソースファイルの登録

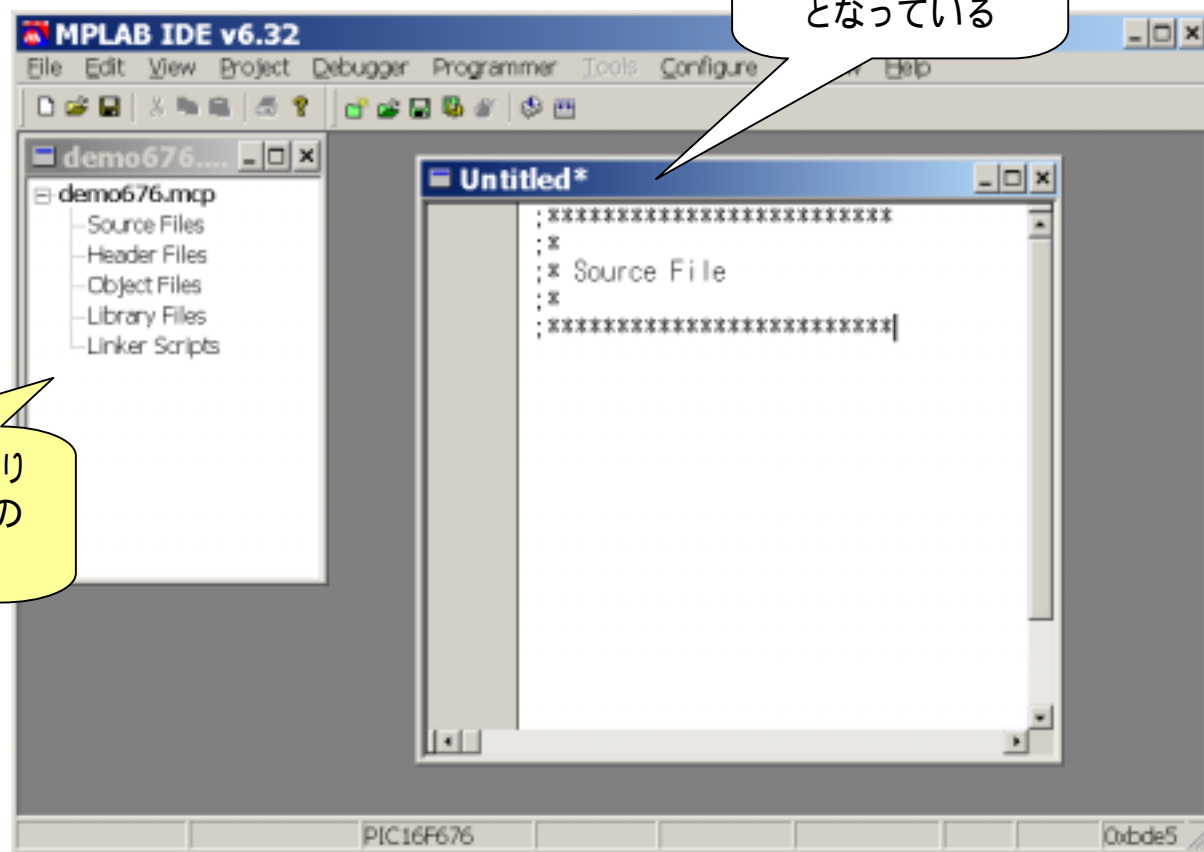


ソースファイルの作成

- ◆ エディタを起動
- ◆ 保存
- ◆ プロジェクトに登録

ソースファイルの作成

- ◆ File New で新規ソースファイルの作成開始
 - ◆ 通常のテキストエディタと同じ扱い
 - ◆ 既存ファイルからのコピーも可能



エディタの設定

◆ 文字フォントの設定

- ◆ Edit Properties の下記ダイアログで Text タグ 選択
- ◆ Select Font ボタンを押してフォントを指定する
- ◆ 日本語フォントを指定すれば日本語コメント表示可能

◆ TAB サイズも設定

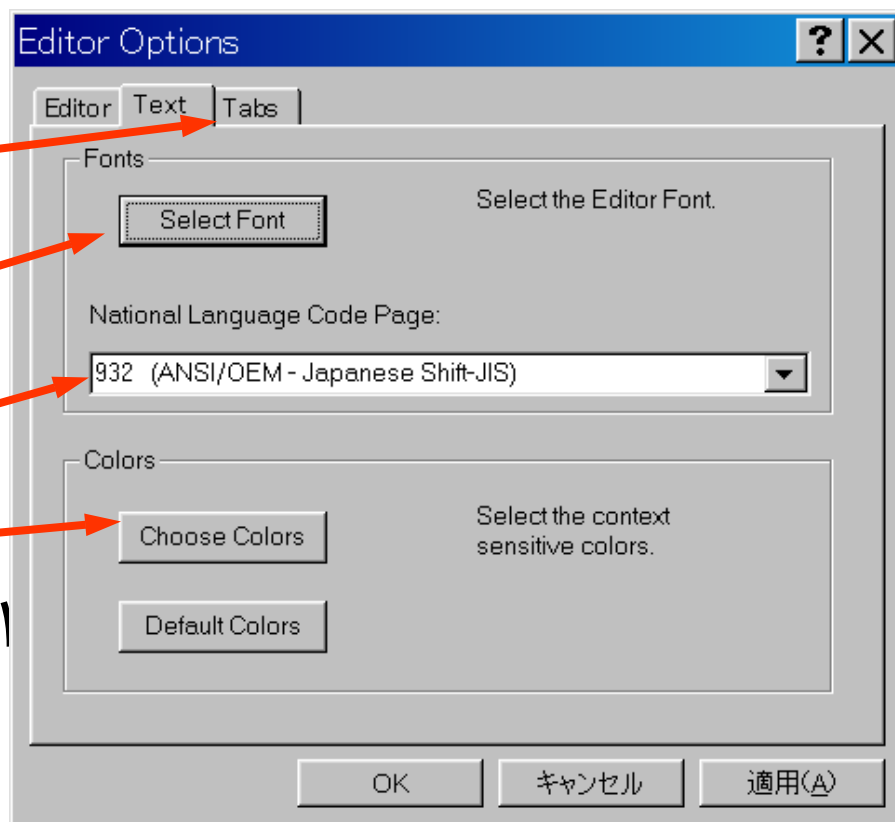
- ◆ 8 文字が良い

◆ フォントの設定

- ◆ 日本語フォント指定

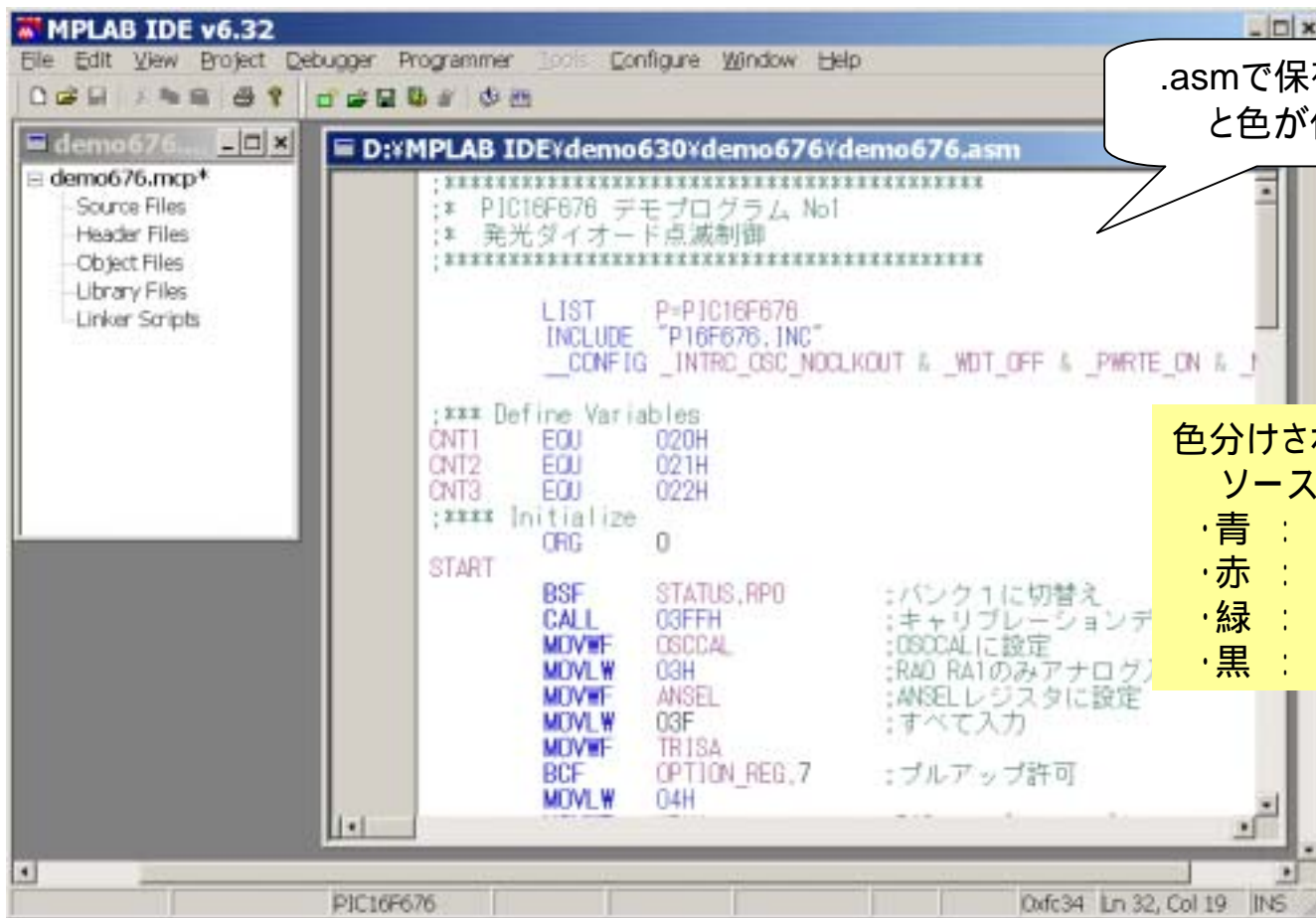
◆ 表示色の設定も可能

- ◆ デフォルトのままでも良い



ソースファイルの作成と保存

- ◆ ソースファイルを作成後File Save as で保存
 - ◆ プロジェクトのフォルダ内に保存すること
 - ◆ 名称は、「FileName.asm」で拡張子は.asm



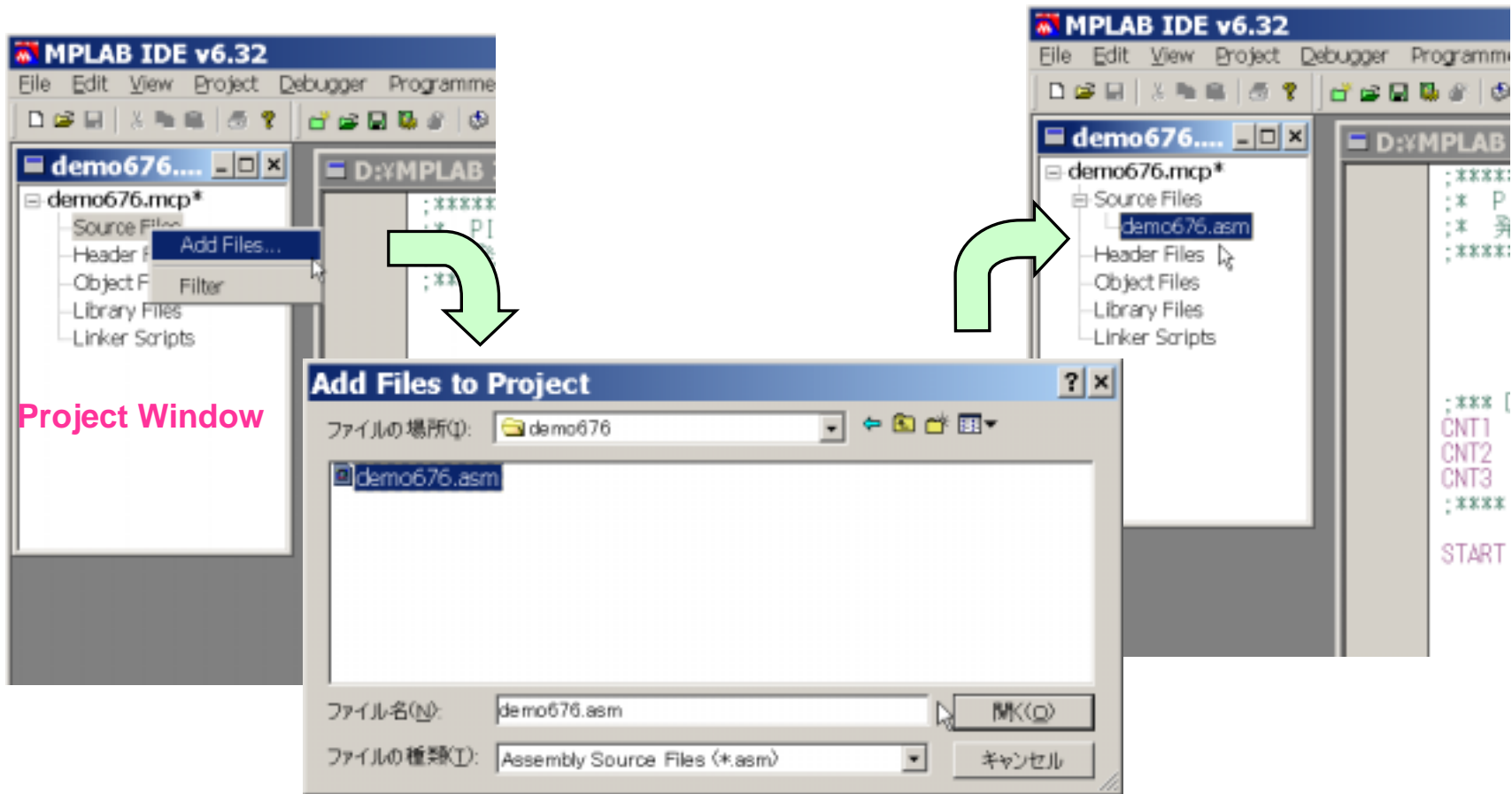
.asmで保存する
と色が付く

色分けされた
ソースファイルリスト

- ・青 : 予約語
- ・赤 : ラベル
- ・緑 : コメント
- ・黒 : リテラル定数

ソースファイルの登録

- ◆ プロジェクトに新規ソースファイルを登録する
 - ◆ Project Window の Source File で右クリック
 - ◆ Add Files でファイルを指定して登録



アセンブル実行

- ◆ Buid Allで実行
- ◆ エラー修正後再Build

アセンブル

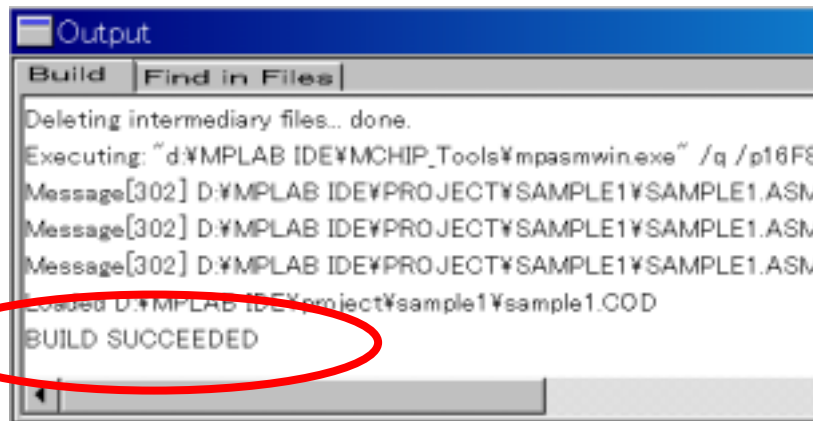
◆ Project Build All

- ◆ **アセンブル**がはじまり、その結果がOutput窓に報告されます。
- ◆ 正常完了なら、BUILD SUCCEEDED
- ◆ 正常完了すると、**HEXファイル**が生成される

アセンブル経過表示



アセンブル結果表示(正常終了の場合)



アセンブルエラーの場合

◆ メッセージは3種類

(1) Message

注意メッセージなので正常

(2) Warning

間違いではないが、直すべき箇所

(3) Error

必ず直さなければならない間違い

エラーの経過表示

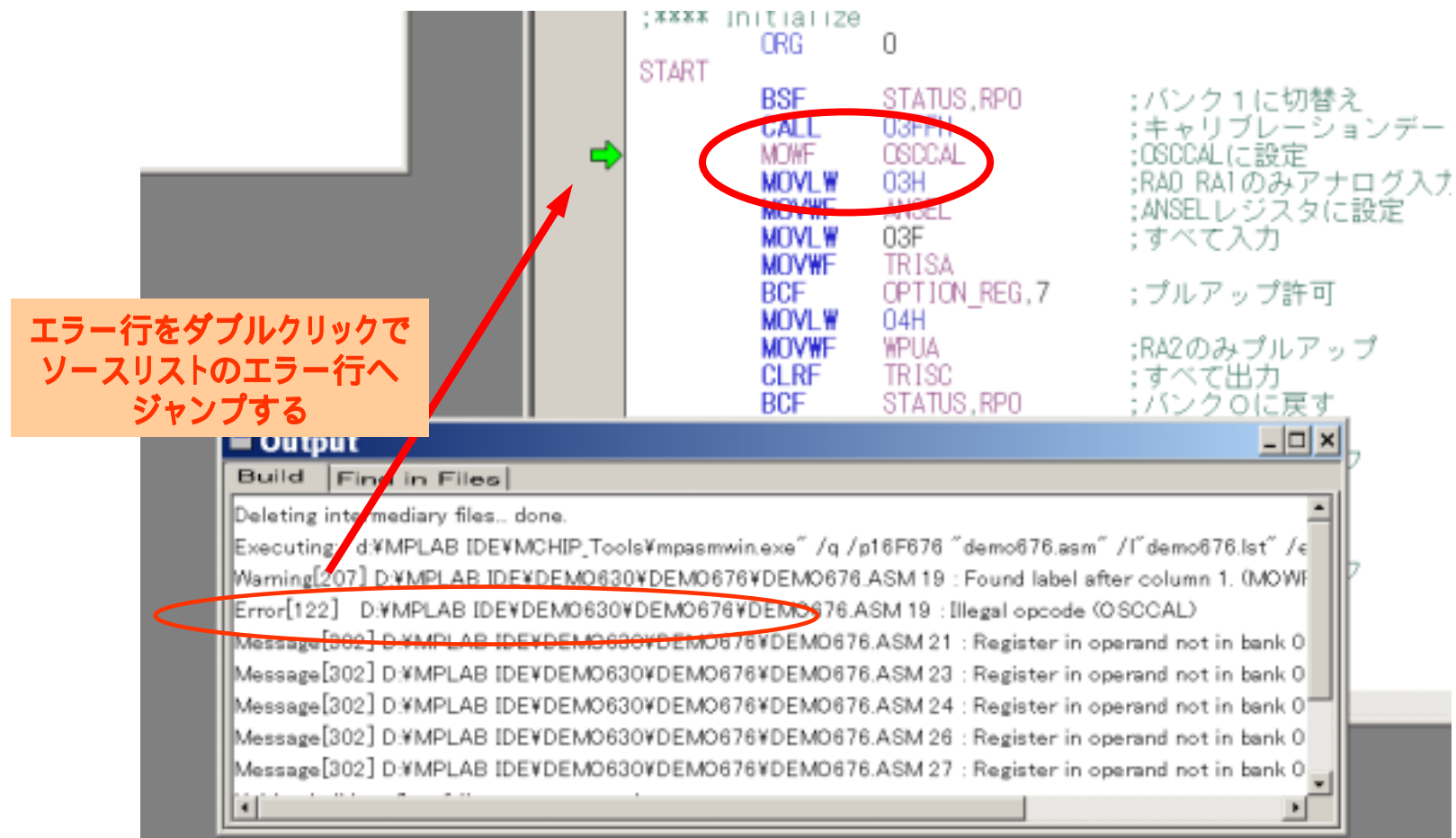


◆ WarningとErrorを全て修正しBuildやり直し

アセンブルエラーの場合

◆ Output窓にError内容表示

- ◆ エラー行ダブルクリックでソースファイルの該当個所にジャンプ
- ◆ 原因追求し修正後、再ビルド

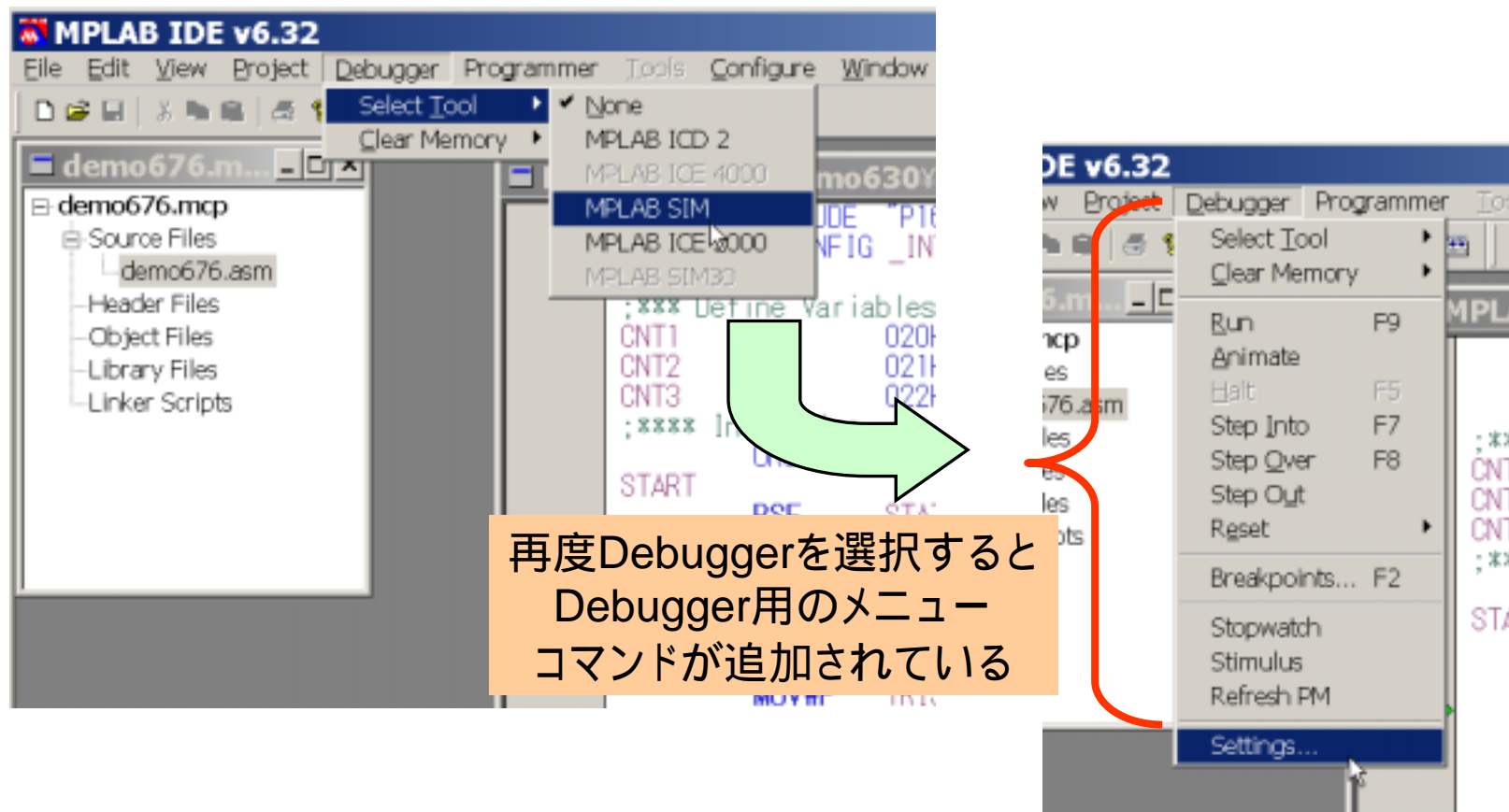


シミュレーションデバッグ

- ◆ シミュレータの選択
- ◆ クロック周波数設定
- ◆ コンパイルリストの表示
- ◆ ブレークポイントの設定
- ◆ Run実行、Step実行
- ◆ 変数の値の確認
- ◆ ポート入力 of シミュレート
- ◆ 実行時間の測定
- ◆ 特殊機能レジスタの表示

シミュレータの選択設定

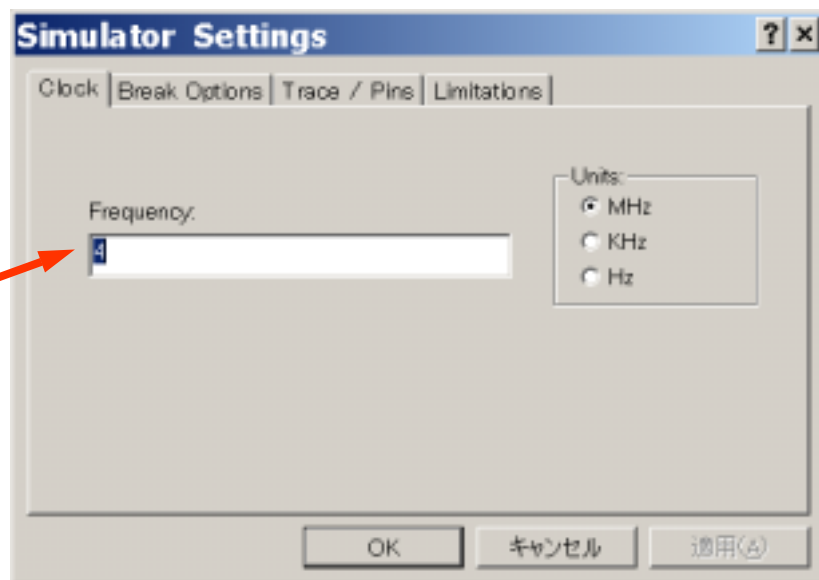
- ◆ Debugger Select Tool MPLAB SIMを選択
 - ◆ シミュレータ用のコマンドが追加される



クロックの周波数設定

- ◆ Debugger Settings で Frequency を設定
 - ◆ シミュレーションの実行時間測定値が正しくなる

4 MHzを指定
(デフォルト)



シミュレーションデバッグの手順

必要な窓を開く

- ◆ ソースリストまたはコンパイルリスト
- ◆ Watch窓で必要なレジスタや変数の表示

ブレークポイントを設定する

- ◆ 最初から順次一旦停止させながらRUNさせる
- ◆ 割込み処理の入りでブレーク

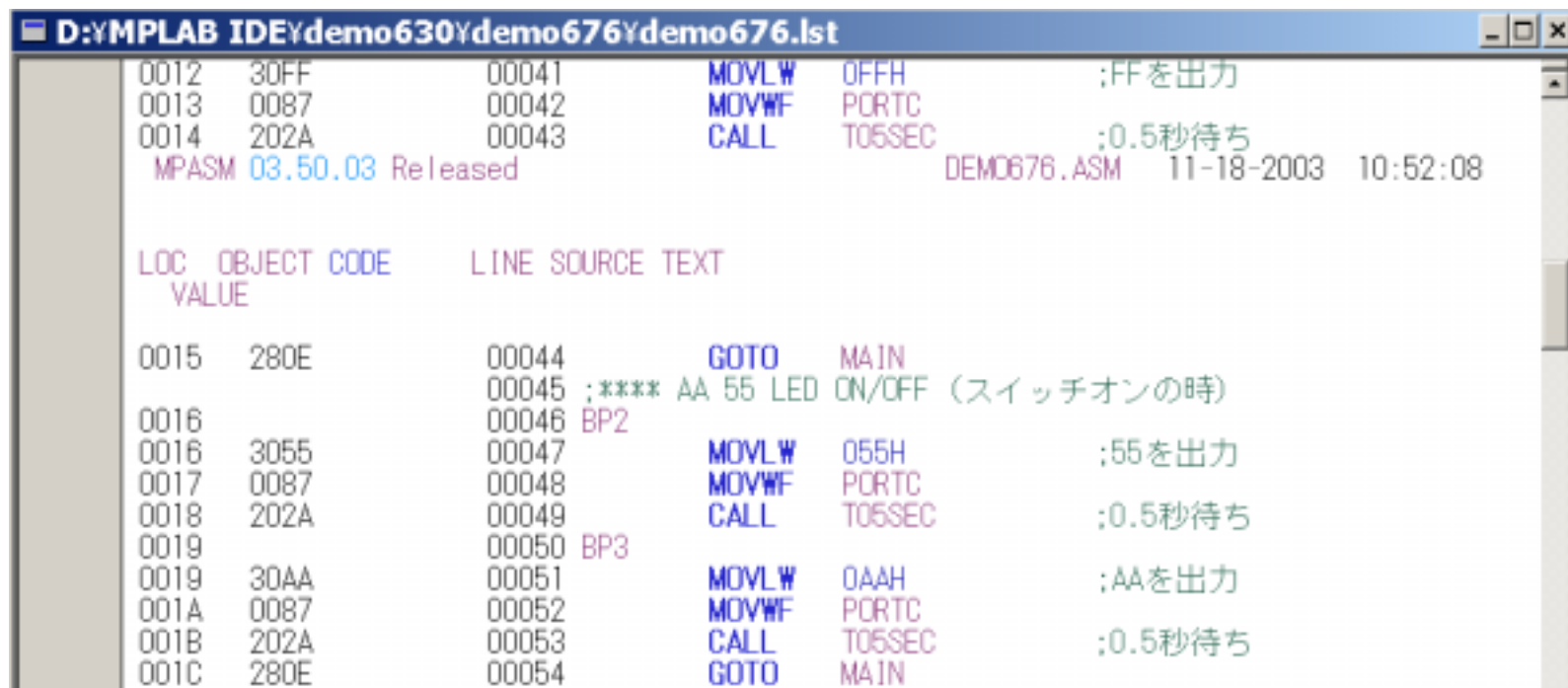
Run、Step into、Step Over で動作実行

- ◆ 各表示窓で動作確認
- ◆ 各レジスタや変数の内容の確認
- ◆ STEP実行による判断命令の確認
- ◆ 実行時間の確認 (Stop Watch)

割込み動作の確認と時間の確認

コンパイルリストの表示

◆ File Open *.lst のファイルを選択



The screenshot shows the MPLAB IDE assembly list window for the file D:\MPLAB IDE\demo630\demo676\demo676.lst. The window displays assembly code with columns for address, object code, line number, source code, and text. The code includes instructions like MOVLW, MOVWF, CALL, and GOTO, along with comments in Japanese. A status bar at the bottom indicates the version of MPASM (03.50.03) and the date/time (11-18-2003 10:52:08).

| LOC | OBJECT | CODE | LINE | SOURCE | TEXT |
|---|--------|------|-------|-----------------------------------|----------------|
| VALUE | | | | | |
| 0012 | 30FF | | 00041 | MOVLW | OFFH ;FFを出力 |
| 0013 | 0087 | | 00042 | MOVWF | PORTC |
| 0014 | 202A | | 00043 | CALL | T05SEC ;0.5秒待ち |
| MPASM 03.50.03 Released DEMO676.ASM 11-18-2003 10:52:08 | | | | | |
| 0015 | 280E | | 00044 | GOTO | MAIN |
| | | | 00045 | ;**** AA 55 LED ON/OFF (スイッチオンの時) | |
| 0016 | | | 00046 | BP2 | |
| 0016 | 3055 | | 00047 | MOVLW | 055H ;55を出力 |
| 0017 | 0087 | | 00048 | MOVWF | PORTC |
| 0018 | 202A | | 00049 | CALL | T05SEC ;0.5秒待ち |
| 0019 | | | 00050 | BP3 | |
| 0019 | 30AA | | 00051 | MOVLW | 0AAH ;AAを出力 |
| 001A | 0087 | | 00052 | MOVWF | PORTC |
| 001B | 202A | | 00053 | CALL | T05SEC ;0.5秒待ち |
| 001C | 280E | | 00054 | GOTO | MAIN |

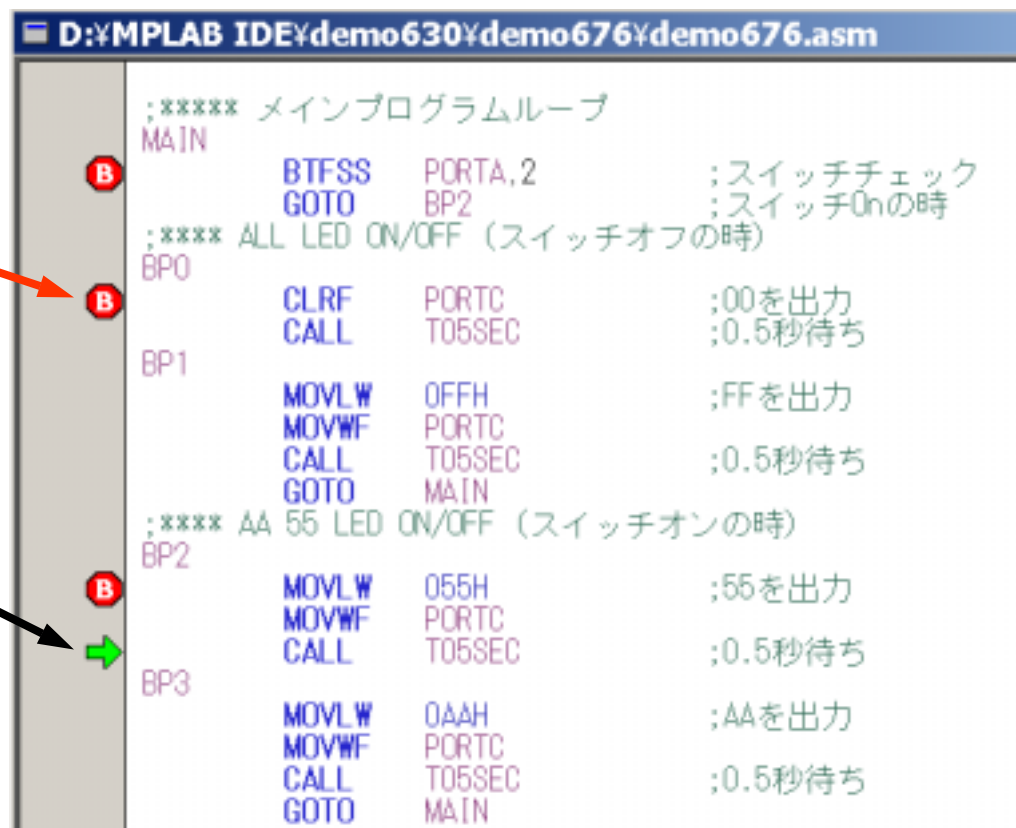
アドレス 機械語 行 ラベル 命令 オペランド ;コメント(日本語OK)

ブレークポイントの設定

- ◆ ソースファイルの先頭でダブルクリック
 - ◆ ポート出力、変数データの内容確認
 - ◆ ポート入力待ちへの擬似入力
 - ◆ STEP実行で分岐方向確認

ダブルクリックでBP設定
再度ダブルクリックで解除

現在実行停止行
STEP実行で進む



```

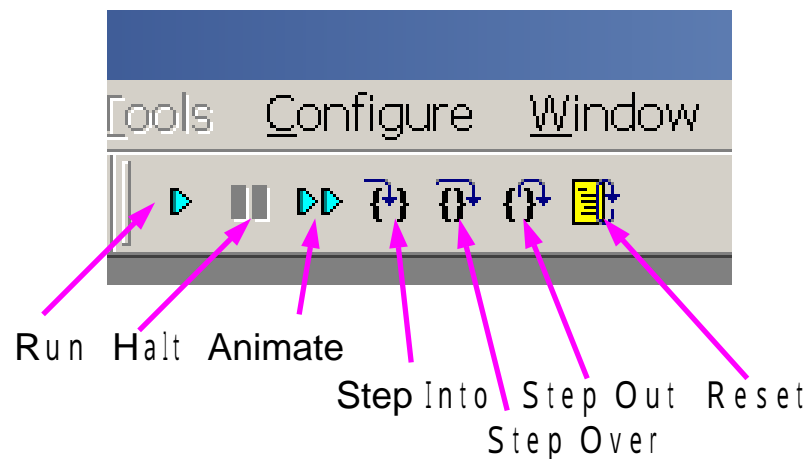
;***** メインプログラムループ
MAIN      BTFSS    PORTA,2      ;スイッチチェック
          GOTO     BP2          ;スイッチOnの時
;**** ALL LED ON/OFF (スイッチオフの時)
BP0       CLRF     PORTC        ;00を出力
          CALL     T05SEC       ;0.5秒待ち
BP1       MOVLW    0FFH         ;FFを出力
          MOVWF    PORTC
          CALL     T05SEC       ;0.5秒待ち
          GOTO     MAIN
;**** AA 55 LED ON/OFF (スイッチオンの時)
BP2       MOVLW    055H         ;55を出力
          MOVWF    PORTC
          CALL     T05SEC       ;0.5秒待ち
BP3       MOVLW    0AAH         ;AAを出力
          MOVWF    PORTC
          CALL     T05SEC       ;0.5秒待ち
          GOTO     MAIN

```

プログラムの実行制御

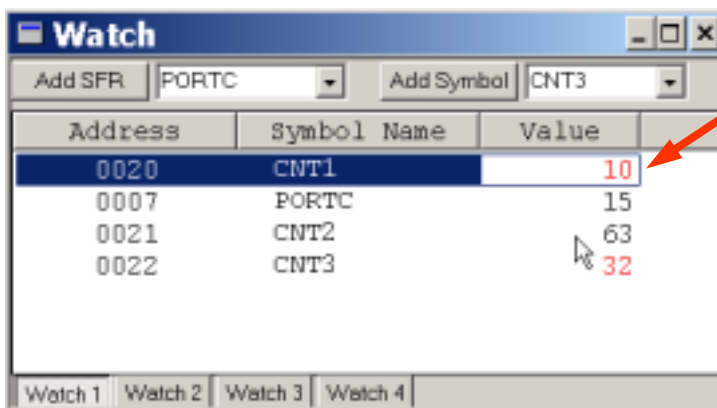
アイコンでも可能

- ◆ Run (**F9**キー)
 - ◆ 現在位置から実行開始
- ◆ Halt (**F5**キー)
 - ◆ 強制的に停止させる
- ◆ Step Into (**F7**キー)
 - ◆ 命令を1個ずつ実行する
- ◆ Step Over (**F8**キー)
 - ◆ サブルーチンの内部にはStepで入らないでRun
 - ◆ 実行時間のかかるサブルーチンは効率良く実行できる
- ◆ Step Out
 - ◆ 一度入ったサブルーチンをRunで抜ける
- ◆ Reset (**F6**キー)
 - ◆ 最初に一度リセットし初期化する



変数の値の確認方法

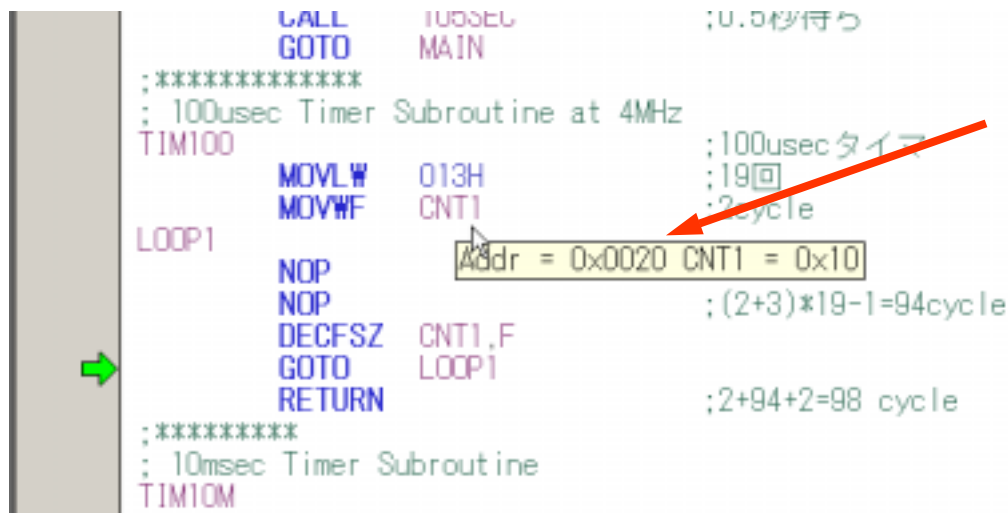
- ◆ View Watch で常時値を確認できる
 - ◆ Add FSRでレジスタの表示追加
 - ◆ Add Symbol ボタンで変数の表示追加



BPで停止したとき
前回との間で変化が
あったものは赤字表示

変数の値の確認方法

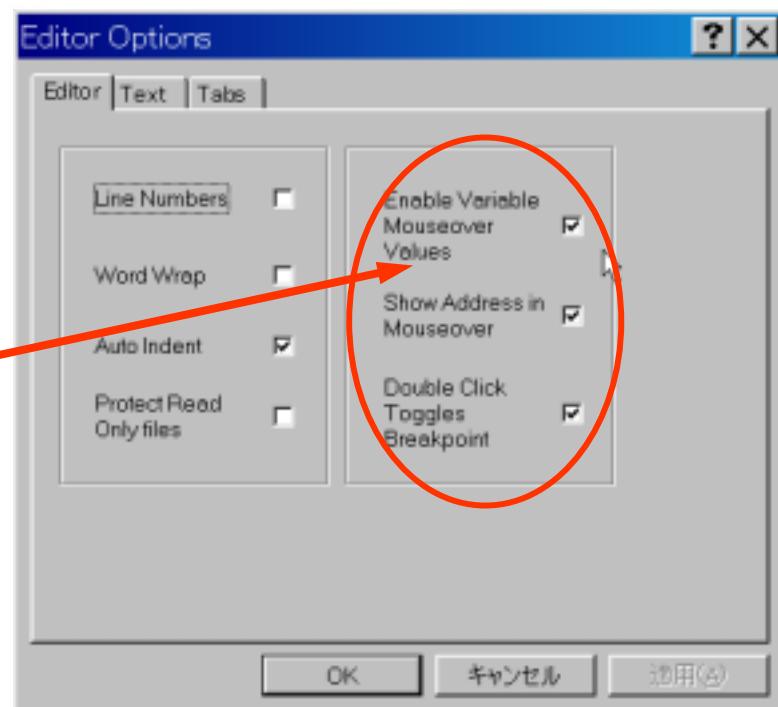
◆ ソースリストで直接確認



マウスカーソルを置けば
その変数の現在値と
アドレスを表示

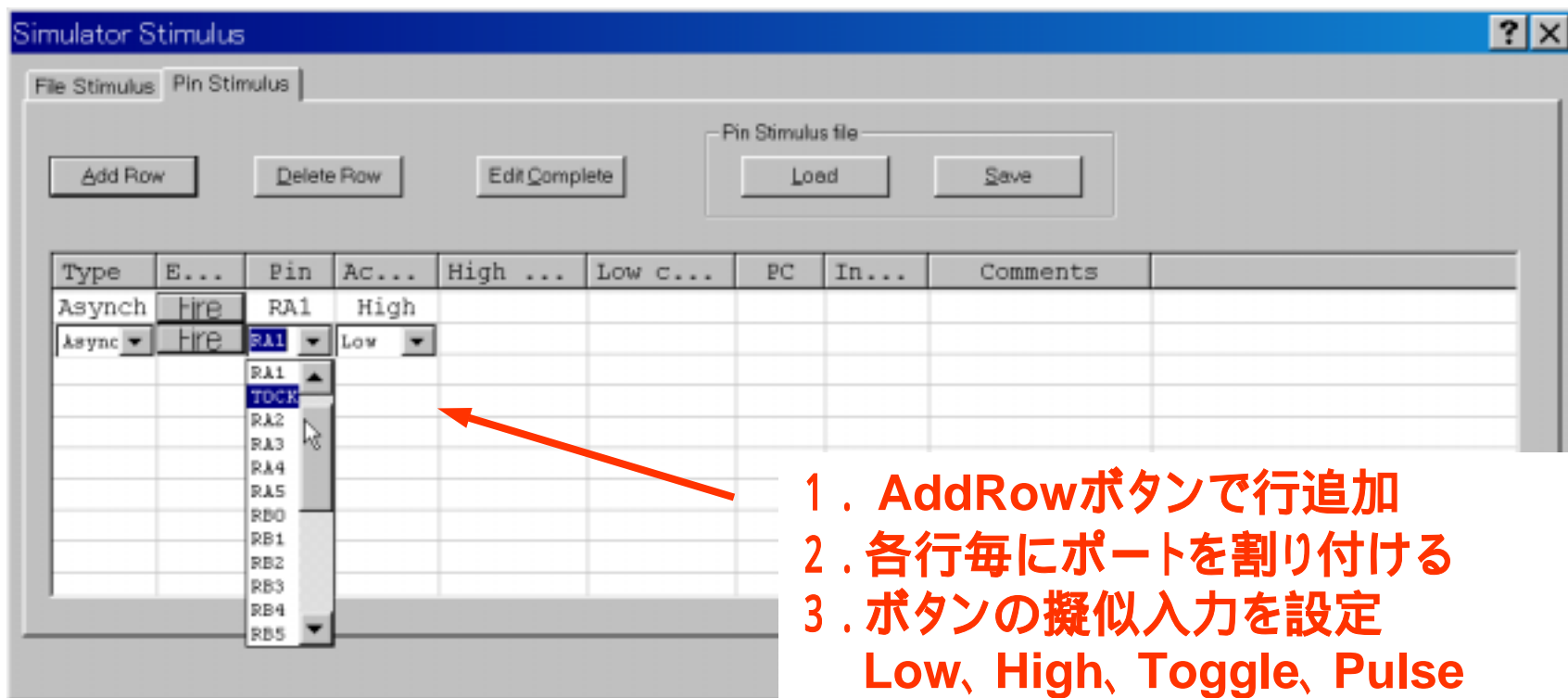
◆ 表示させるための設定

◆ Edit Properties で
チェック追加



ポートの擬似入力

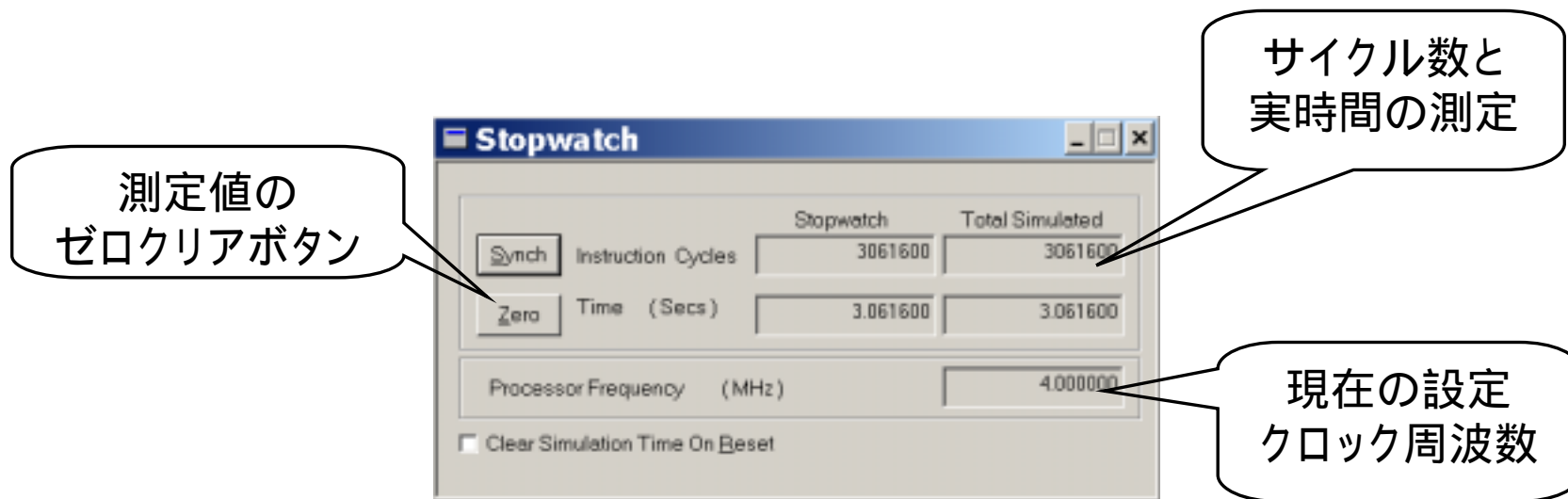
- ◆ Debugger Stimulus Pin Stimulus で設定し入力
 - ◆ 入力はFireボタンを押すことでいつでも可能



1. AddRowボタンで行追加
2. 各行毎にポートを割り付ける
3. ボタンの擬似入力を設定
Low、High、Toggle、Pulse
4. ブレークさせてFireボタンを押す

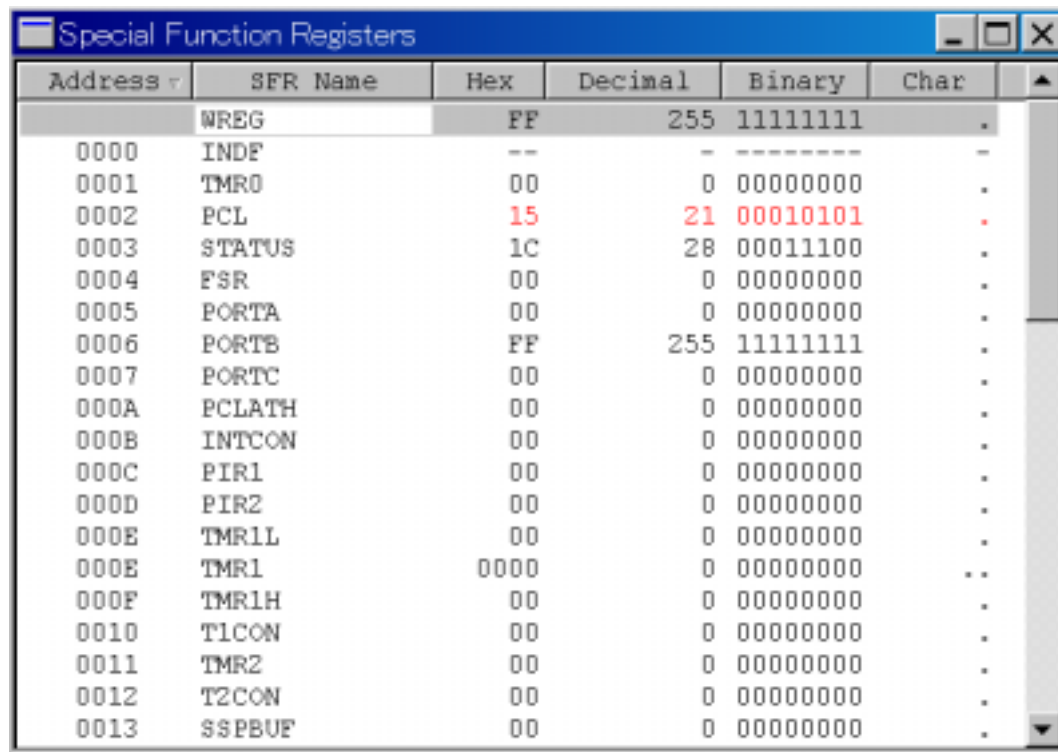
実行時間の測定方法

- ◆ Debugger Stopwatch
 - ◆ ソフトウェアタイマの作成、確認に便利
 - ◆ 実際のプログラム実行時間を測定できる
 - ◆ システムクロックの周波数設定が必要
(Debug Settings)



特殊機能レジスタの表示

- ◆ View Special Function Registers
 - ◆ STATUSレジスタやPORTレジスタの現在地を表示
 - ◆ 前回表示時点から変更があったものを赤字で表示
 - ◆ ブレークポイントなどで停止したときに表示する



| Address | SFR Name | Hex | Decimal | Binary | Char |
|---------|----------|------|---------|----------|------|
| | WREG | FF | 255 | 11111111 | . |
| 0000 | INDF | -- | - | ----- | - |
| 0001 | TMR0 | 00 | 0 | 00000000 | . |
| 0002 | PCL | 15 | 21 | 00010101 | . |
| 0003 | STATUS | 1C | 28 | 00011100 | . |
| 0004 | FSR | 00 | 0 | 00000000 | . |
| 0005 | PORTA | 00 | 0 | 00000000 | . |
| 0006 | PORTB | FF | 255 | 11111111 | . |
| 0007 | PORTC | 00 | 0 | 00000000 | . |
| 000A | PCLATH | 00 | 0 | 00000000 | . |
| 000B | INTCON | 00 | 0 | 00000000 | . |
| 000C | PIR1 | 00 | 0 | 00000000 | . |
| 000D | PIR2 | 00 | 0 | 00000000 | . |
| 000E | TMR1L | 00 | 0 | 00000000 | . |
| 000E | TMR1 | 0000 | 0 | 00000000 | .. |
| 000F | TMR1H | 00 | 0 | 00000000 | . |
| 0010 | T1CON | 00 | 0 | 00000000 | . |
| 0011 | TMR2 | 00 | 0 | 00000000 | . |
| 0012 | T2CON | 00 | 0 | 00000000 | . |
| 0013 | SSPBUF | 00 | 0 | 00000000 | . |

プログラムの書き込み (MPLAB ICD2の場合)

- ◆MPLAB ICD2を接続
- ◆MPLABからプログラムを起動
- ◆書き込み実行

MPLAB ICD2の接続

- ◆ MPLAB ICD2のUSBコネクタを接続
- ◆ ICSPケーブルをターゲットのボードに接続

必要に応じて
ACアダプタを接続



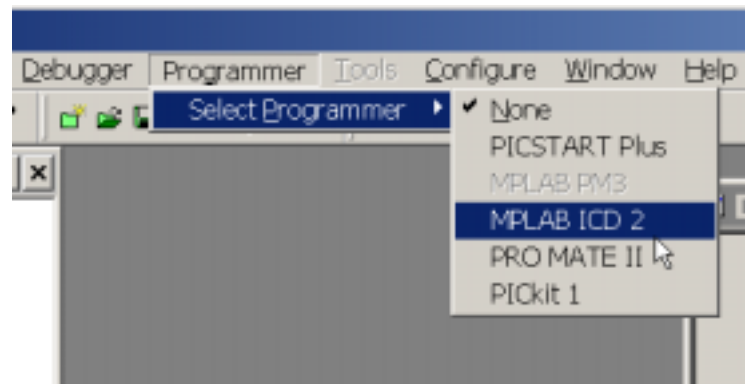
パソコンの
USBへ接続

ターゲット基板の
ICSPコネクタに接続

MPLAB ICD2の接続

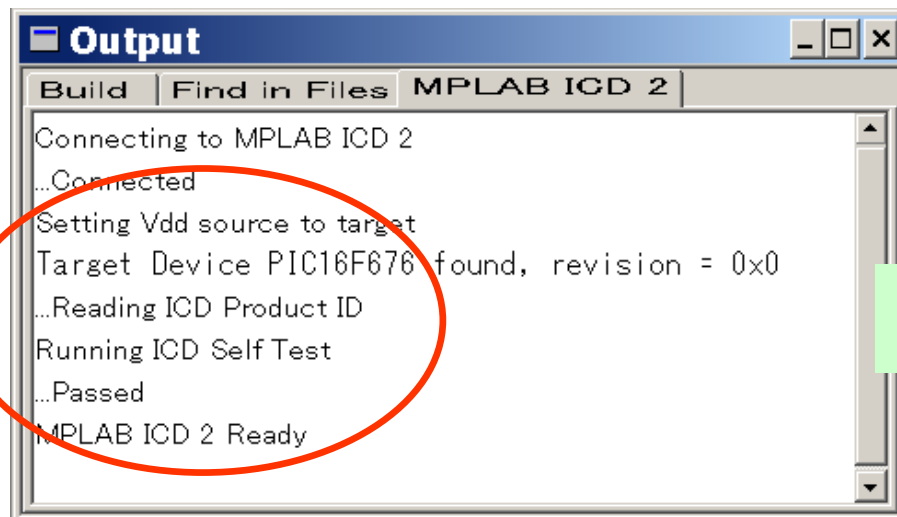
◆ Programmer

Select Programmer
MPLAB ICD2

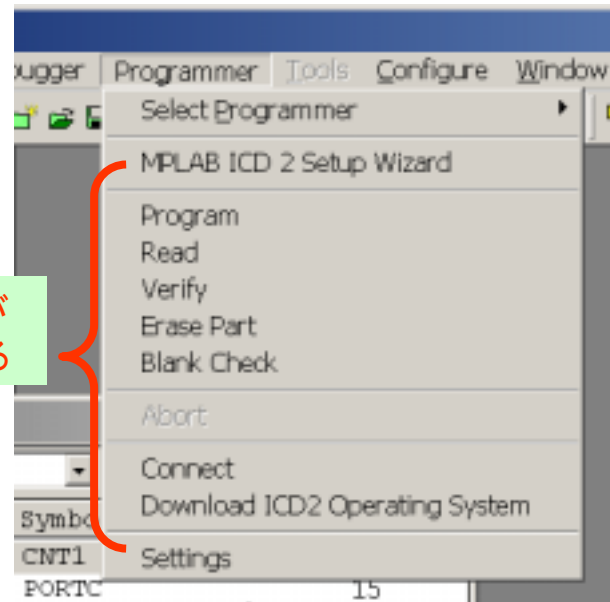


◆ Programmer Enable Programmer

◆ 正常であればメッセージが表示されメニューが追加されます。



メニューが
追加される



電源供給方法の設定

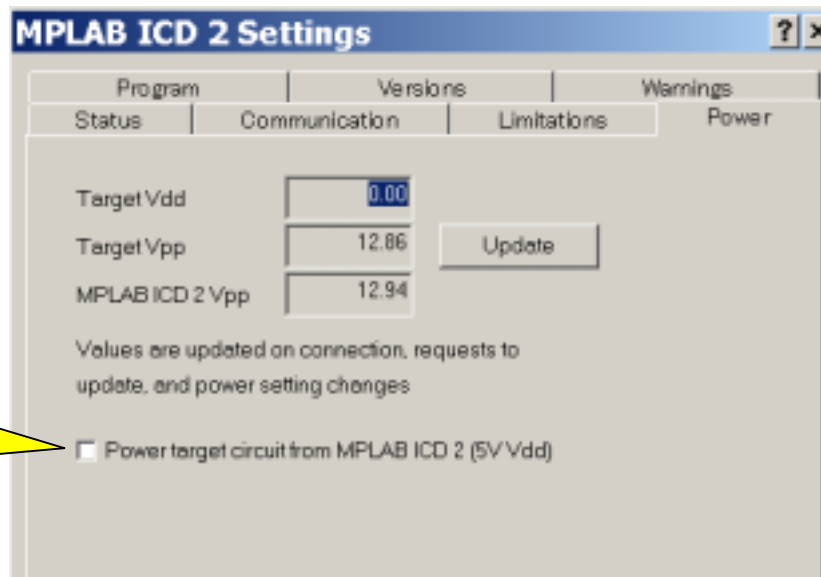
◆ Programmer Settings Powerタブで設定

◆ 通常のICSPの時

→ ターゲットからVdd供給とする

◆ PIC単独でプログラムする時

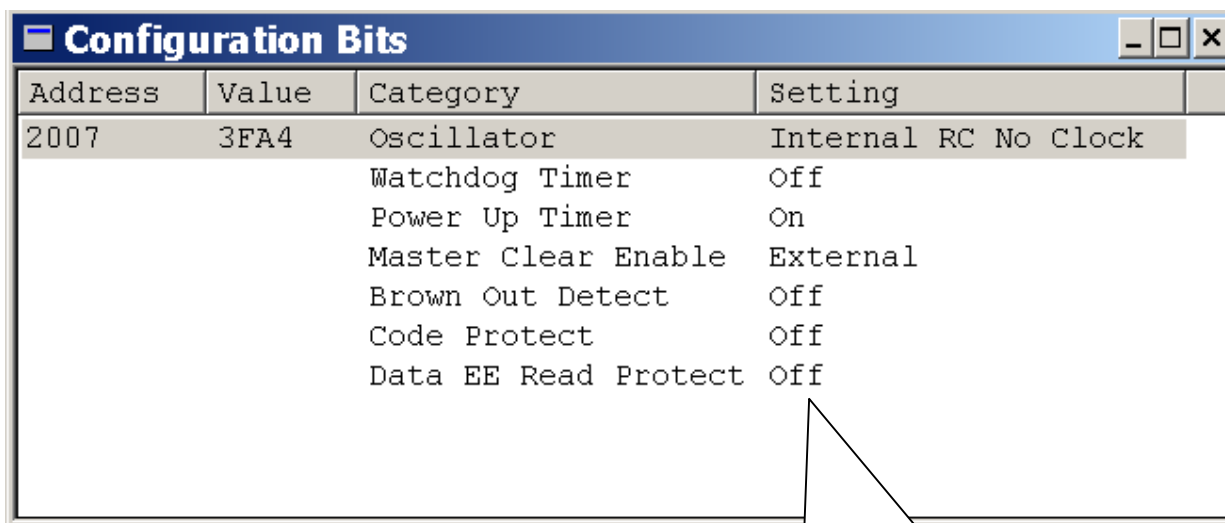
→ ICD2からVddを供給



このチェックを
はずすとICD2か
らVdd供給となる

コンフィギュレーションの確認

- ◆ Configure Configuration Bits
下記ダイアログで内容を確認する。



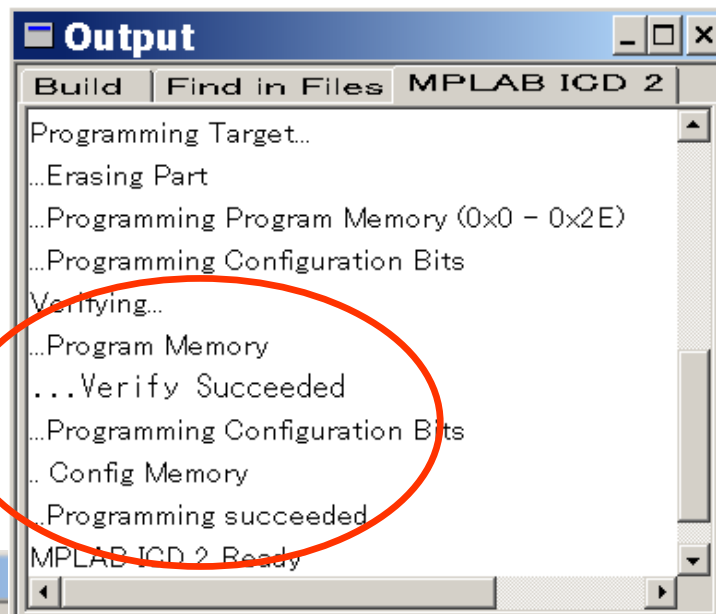
| Address | Value | Category | Setting |
|---------|-------|----------------------|----------------------|
| 2007 | 3FA4 | Oscillator | Internal RC No Clock |
| | | Watchdog Timer | Off |
| | | Power Up Timer | On |
| | | Master Clear Enable | External |
| | | Brown Out Detect | Off |
| | | Code Protect | Off |
| | | Data EE Read Protect | Off |

現在の設定値を確認
変更の場合には、
ここで選択肢が表示される
ので、そこで指定

書き込み実行

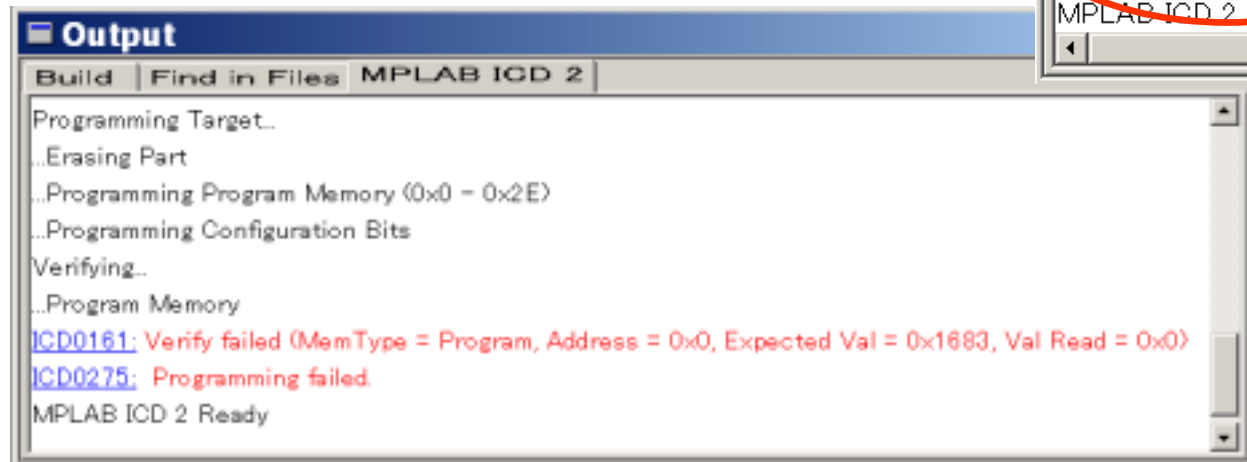
- ◆ Programmer Program
で書き込み実行開始
- ◆ 正常完了ならICSPケーブル
を抜けば実行開始する

正常完了結果の表示



この表示を出すには
View Output

異常結果の表示



終わり

◆ありがとうございました！

プログラムの書込み (PICSTART Plusの場合)

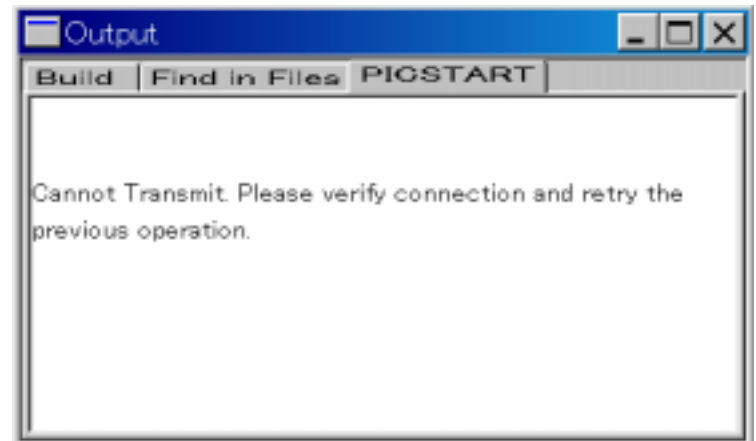
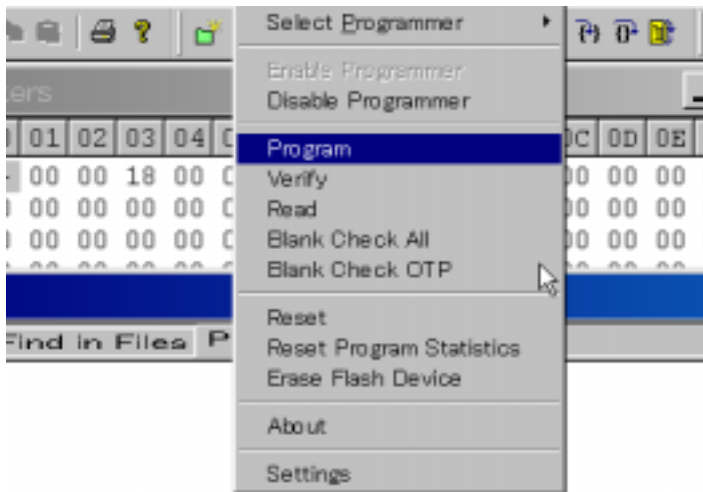
- ◆PIC START Plusを接続
- ◆MPLABからプログラムを起動
- ◆書き込み実行

PICSTART Plus

- ◆ PICSTART PlusにACアダプタを接続して、電源を入れます。
- ◆ パソコンのシリアル・ポートとケーブルで接続します。
 - ◆ COMポート番号を確認しておきます。
(普通は自動認識されます)

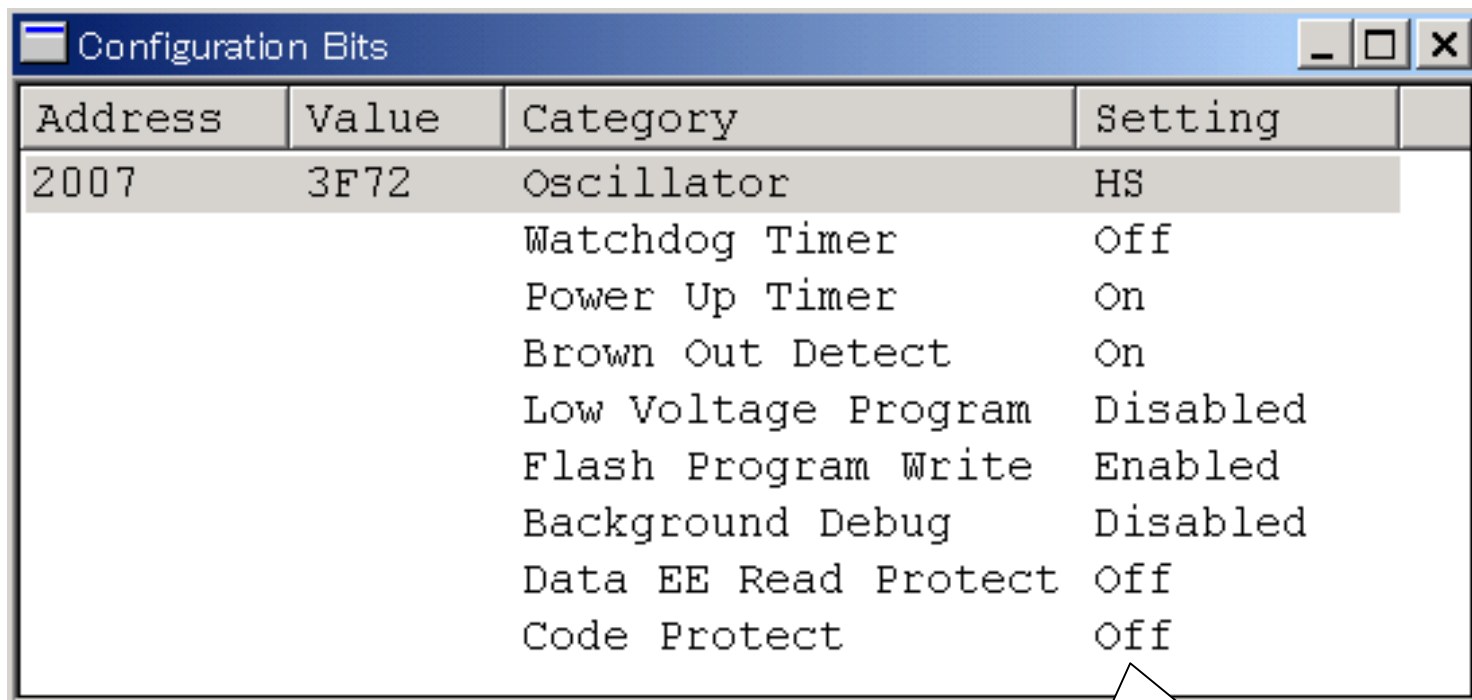
PICSTART Plus

- ◆ Programmer Select Programmer PICSTART Plus
- ◆ Programmer Enable Programmer
 - ◆ 正常であれば左下図のようにProgrammerのメニューに項目が追加されます。
 - ◆ 異常なときは右下図のようなメッセージが表示されます。



コンフィギュレーションの確認

- ◆ Configure Configuration Bits
で開く下記ダイアログで内容を確認する。



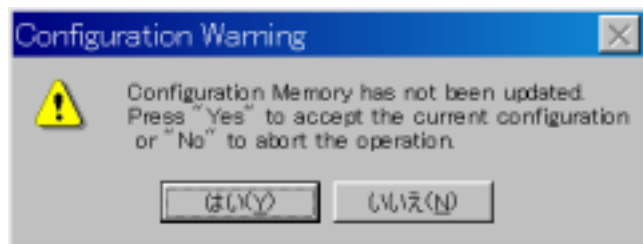
| Address | Value | Category | Setting |
|---------|-------|----------------------|----------|
| 2007 | 3F72 | Oscillator | HS |
| | | Watchdog Timer | Off |
| | | Power Up Timer | On |
| | | Brown Out Detect | On |
| | | Low Voltage Program | Disabled |
| | | Flash Program Write | Enabled |
| | | Background Debug | Disabled |
| | | Data EE Read Protect | Off |
| | | Code Protect | Off |

現在の設定値を確認
変更の場合には、
ここで選択肢が表示される
ので、そこで指定

PICSTART Plus

- ◆ PICをPICSTART Plusのソケットに挿入します。
 - ◆ ProgrammerメニューのProgramをクリックします。
下記Warningメッセージが出ますが「はい」とします。
 - ◆ 書き込み実行中は左下のSTATUS行に進行状況表示
 - ◆ 完了後Outputダイアログを表示していればメッセージで通知

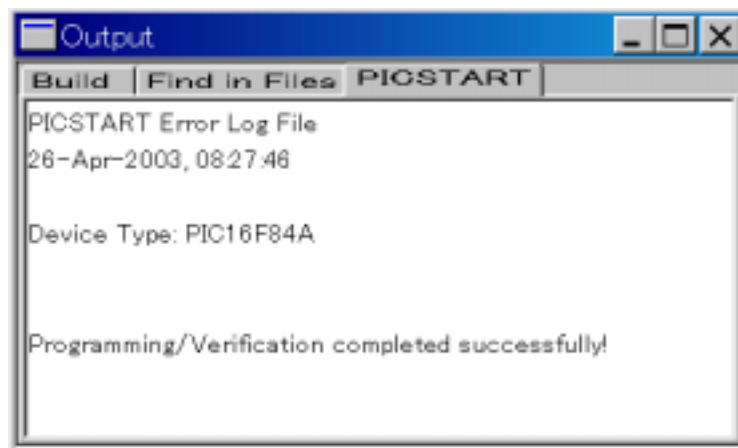
Warningメッセージ



進行状況の表示



結果の表示

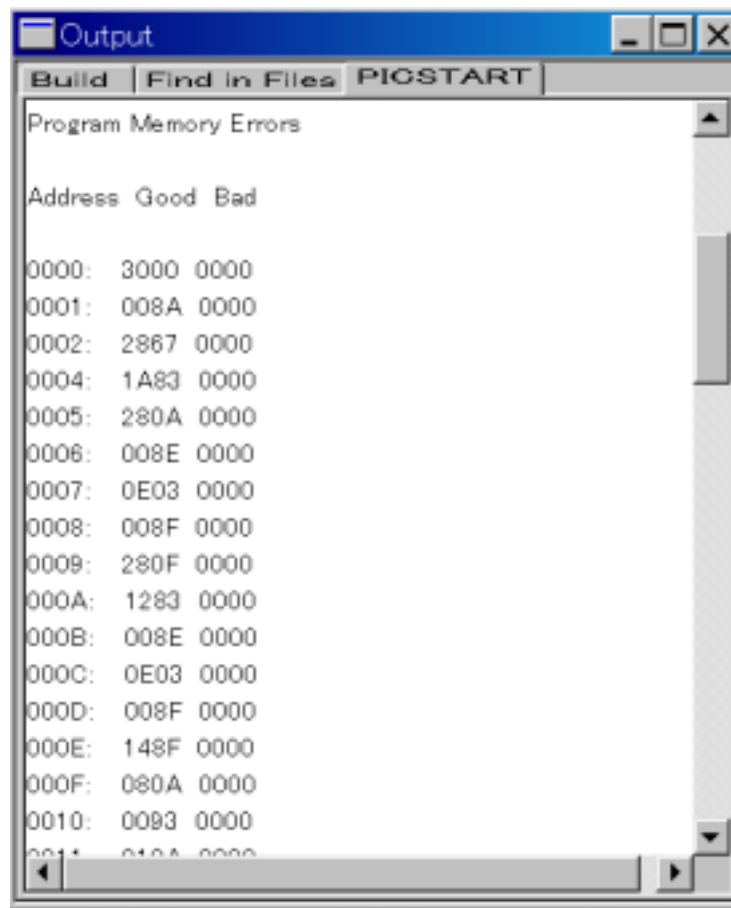


この表示を出すには
View Output

PICSTART Plus

- ◆ エラーがあったときは、Outputメッセージでエラー・ログ・ファイルが開く

エラー発生アドレス
正しい値(Good)
間違った値(Bad)
で一覧表示される



PICの実装

- ◆ 書き込み完了したPICをボードのICソケットに実装する。
 - ◆ 向きを間違えないように
 - ◆ PICのピンを曲げないように
- ◆ 実習用基板の電池を接続する
 - ◆ 正常であればこれで期待通りに動作するはず・・・