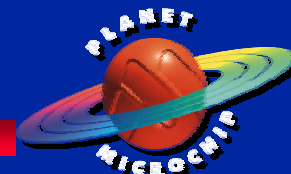


MPASM アセンブラ実習

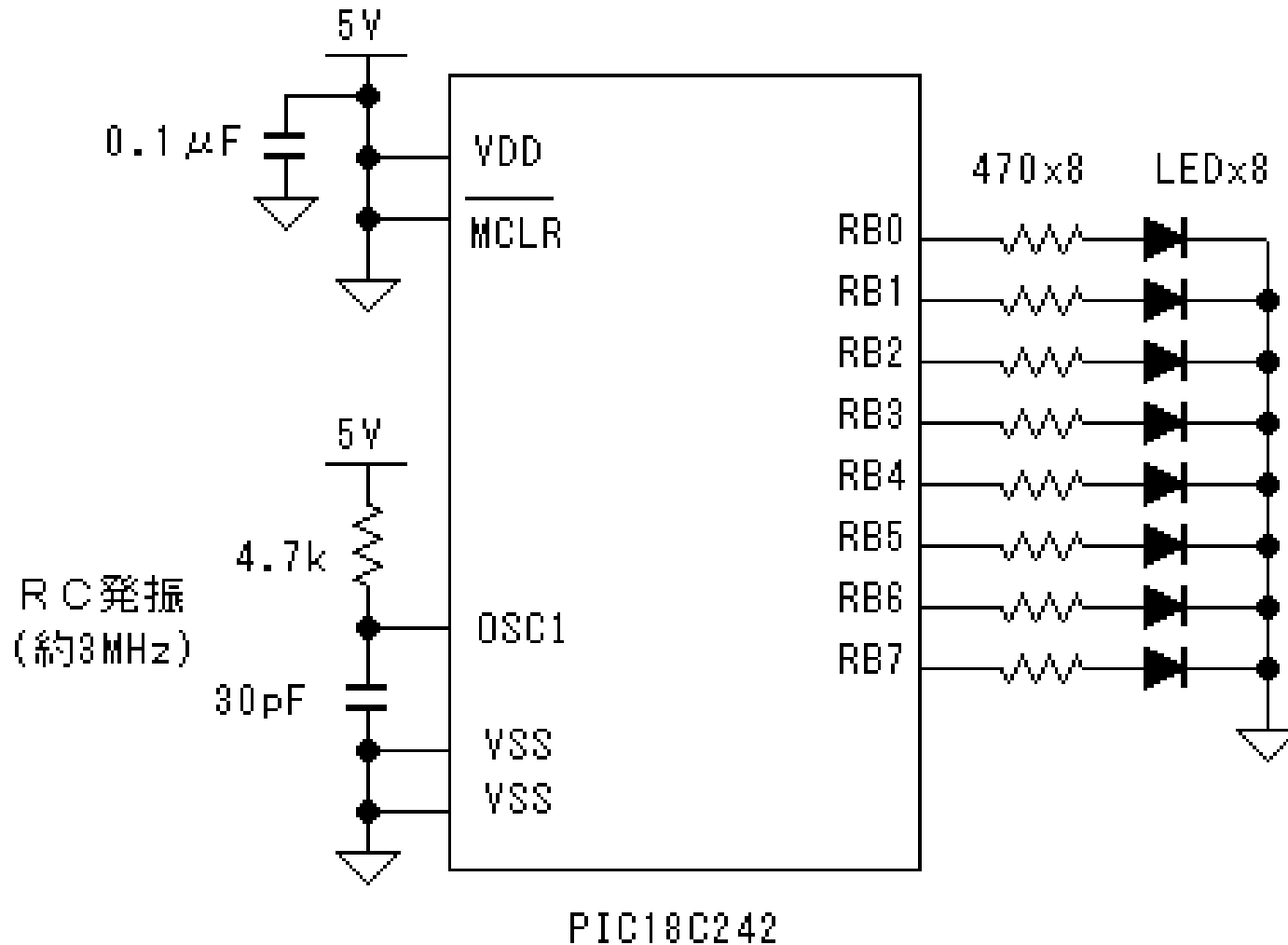


実習内容

- PIC18C242のポートBに接続したLEDを点滅させます(他の実習と同じ内容です)。
 - 8個のLEDに表示される値が+3ずつインクリメントします。
- MPLABを使って、MPASMアセンブラのソース・ファイルのアセンブルします。
 - MPLABとソース・ファイルは配布したCD-Rのなかに入っています。



ブロック図



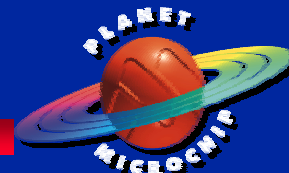
ソース・リスト(1)

```
LIST P=18C242,F=INHX32           ;マイコンはPIC18C242
INCLUDE          P18C242.INC      ;インクルード・ファイルを読み込む

                                ;コンフィギュレーション・ビットの設定
_CONFIG _CONFIG0, _CP_OFF_0
_CONFIG _CONFIG1, _OSCS_OFF_1 & _RCIO_OSC_1
_CONFIG _CONFIG2, _BOR_ON_2 & _BORV_42_2 & _PWRT_ON_2
_CONFIG _CONFIG3, _WDT_ON_3 & _WDTPS_128_3
_CONFIG _CONFIG5, _GCP2MX_ON_5
_CONFIG _CONFIG6, _STVR_ON_6
;コード・プロテクト=OFF,システム・クロック切り替え=OFF,RC発振でOSC2ピンはRA6,
;ウォッチドッグ・タイマ=ON,ポストスケーラ1:128,4.2Vブラウンアウト・リセット,
;パワーアップ・タイマ=ON,GCP2入出力はRC1ピンに割り当て,
;スタック・オーバーフロー/アンダーフロー・リセット=ON

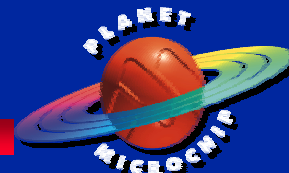
;-----
UPDATE      EQU          7           ;LEDの更新周期
;-----

CBLOCK      0
W_TEMP      ;コンテキスト保存用(WレジスタとSTATUSレジスタ)
STATUS_TEMP
CNT          ;LEDの更新周期用のカウンタ
PORTB_OUT   ;ポートB用の出力バッファ
ENDC        ;CBLOCK終了
;-----
```



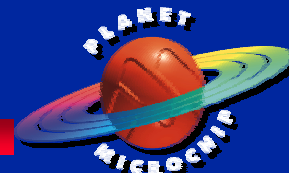
ソース・リスト(2)

```
ORG 0 ;リセット・ベクタ(プログラム・アドレス0番地)
GOTO START ;STARTへ跳ぶ
-----
ISR
ORG 8 ;割り込みベクタ(プログラム・アドレス8番地)
;割り込み処理ルーチン
BTFSSINTCON,T0IF,A ;タイマ0割り込みでなければISR_ENDへ跳ぶ
GOTO ISR_END
BCF INTCON,T0IF,A ;タイマ0割り込みフラグを0にする
DECFSZ CNT,F,A ;CNTを-1して0でなければISR_ENDへ跳ぶ
GOTO ISR_END
MOVLW UPDATE ;CNTにLEDの更新周期を書き込む
MOVWF CNT,A
MOVLW D'3' ;PORTB_OUTの内容を+3する
ADDWF PORTB_OUT,F,A
ISR_END
RETFIE FAST ;コンテキストを復帰して割り込みから復帰
-----
IOPORT ;サブルーチン:I/Oポート設定
CLRF LATA,A ;ポートAの出カラッチに0を書き込む
MOVF PORTB_OUT,W,A ;PORTB_OUTの内容をポートBの出カラッチに書き込む
MOVWF LATB,A
CLRF LATC,A ;ポートCの出カラッチに0を書き込む
CLRF TRISA,A ;ポートAをすべて出力ピンにする
CLRF TRISB,A ;ポートBをすべて出力ピンにする
CLRF TRISC,A ;ポートCをすべて出力ピンにする
RETURN
```



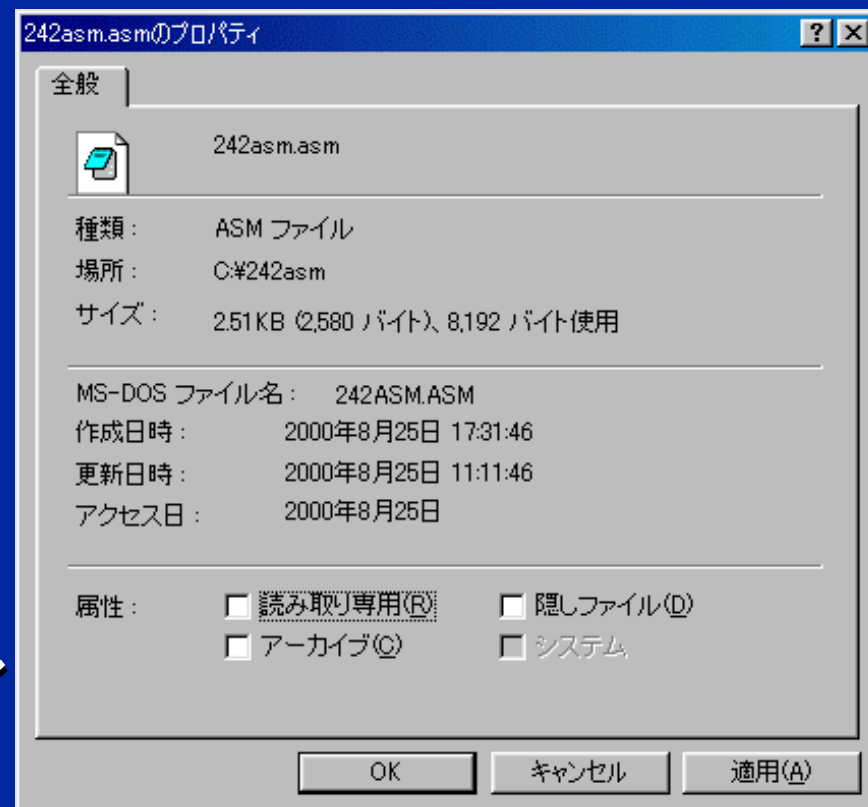
ソース・リスト(3)

```
-----  
START  
    CLRF PORTB_OUT,A           ;PORTB_OUTに0を書き込む  
    CALL IOPORT                ;I/Oポート設定  
    MOVLW    H'D7'             ;タイマ0の設定  
    MOVWF   T0CON,A           ;8ビット・モード、内部クロック、1:256プリスケアラ  
    MOVLW   D'1'              ;CNTに1を書き込む  
    MOVWF   CNT,A            ;  
    CLRF PORTB_OUT,A           ;PORTB_OUTに0を書き込む  
    CLRF TMR0L,A              ;TMRLに0を書き込む  
    MOVLW   H'A0'             ;タイマ0割り込みをイネーブルする  
    MOVWF   INTCON,A          ;  
LOOP  
    CLRWDT                    ;WDTを0にする  
    CALL IOPORT                ;I/Oポート設定  
    GOTO    LOOP              ;LOOPへ跳ぶ  
-----  
END                            ;プログラムの終わり
```



ソース・ファイルのコピー

- 配布CD-Rのなかのsourceフォルダに入っている242asmフォルダをc:¥ドライブにコピーします。
 - 242asmフォルダを開きます。
 - 242asm.asmファイルを右クリックして、メニューからプロパティを選択します。
 - 読み取り専用のチェックを外して、OKをクリックします。



MPLABのインストール

- 配布CD-RのなかのSetupフォルダに入っているMPL51102.exeを起動します。
- 起動後、Nextボタンをクリックし続けます。
 - インストールするフォルダはデフォルトのままにしてください。
 - Enterキーを押すと元に戻るダイアログがあります。
- Finishボタンが表示されたら、そのボタンをクリックしてインストール完了です。
 - C:¥Program Files¥Mplabフォルダにインストールされます。
- 正常にインストールできないときは、MPL51102.exeをハード・ディスクにコピーして、起動してみてください。



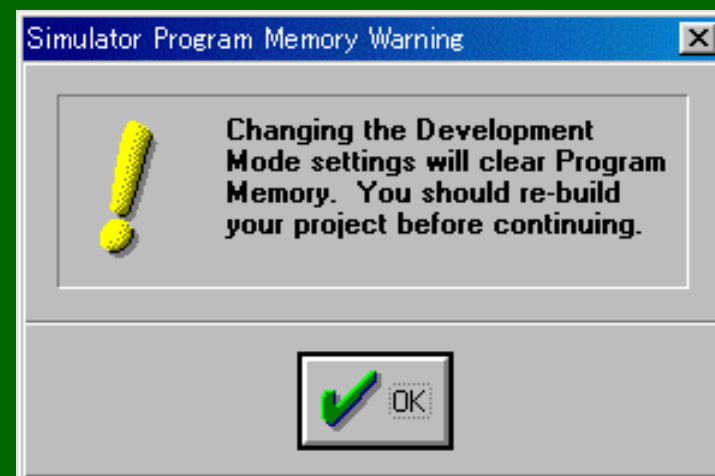
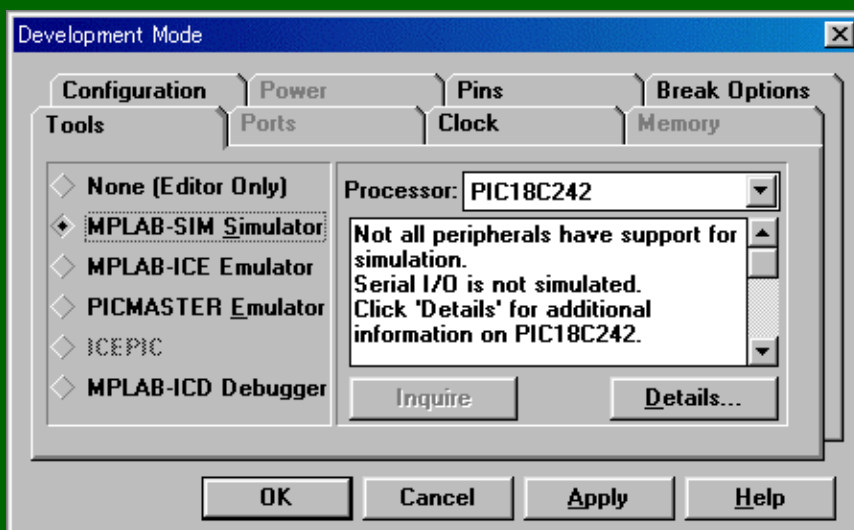
MPLABの起動

- 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→MPLABを起動します。



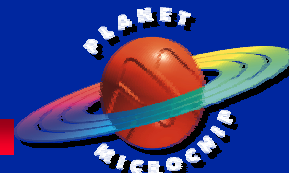
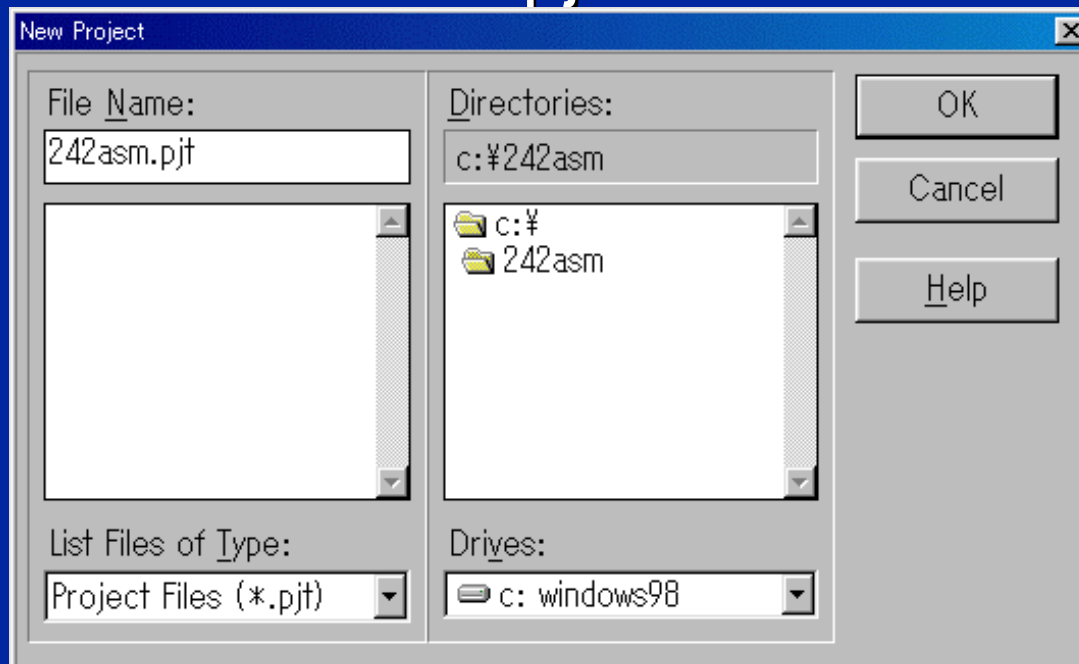
デベロップメント・モードの設定

- Options→Development Modeを選択します。
- 左下図の設定にして、OKをクリックします。
 - Processor:PIC18C242
 - MPLAB-SIM Simulatorをオン
- 右下図が表示されたらOKをクリックします。



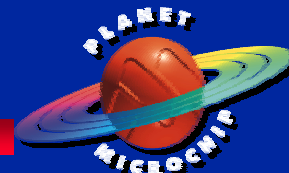
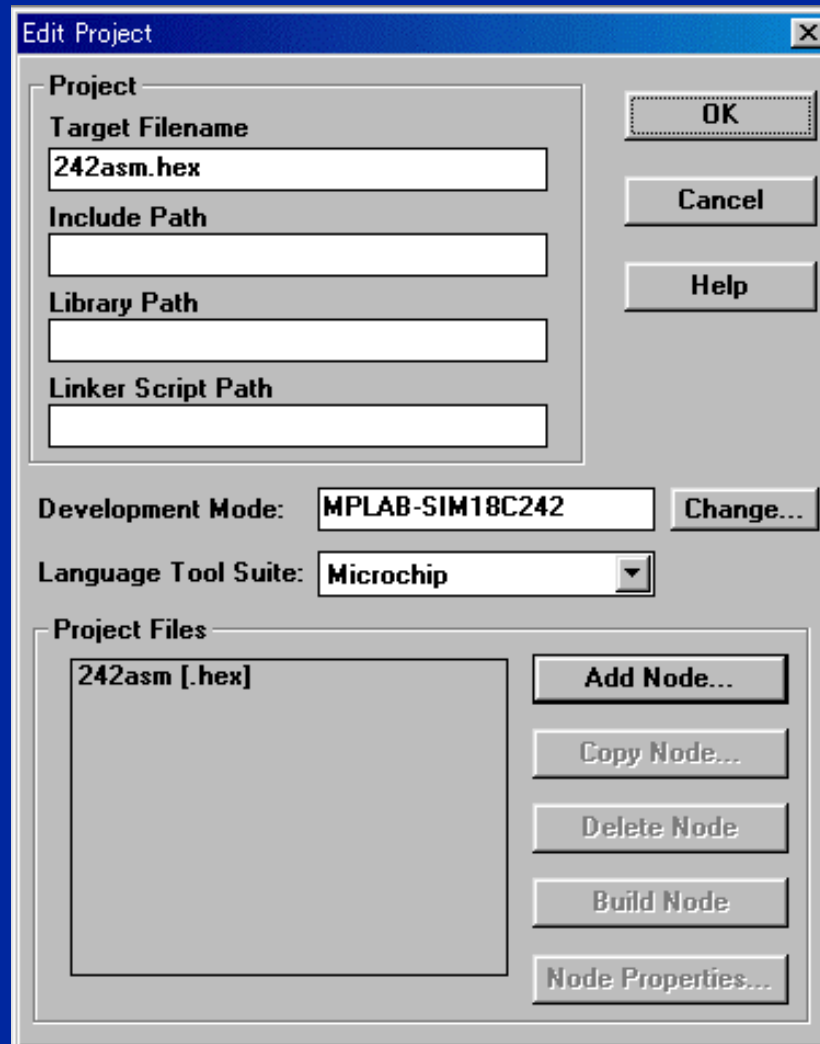
プロジェクトの設定(1)

- Project→New Projectを選択します。
- 以下の設定にして、OKをクリックします。
 - Directories: c:¥242asm
(c:¥をダブルクリック後、242asmをダブルクリック)
 - File Name: 242asm.pjt



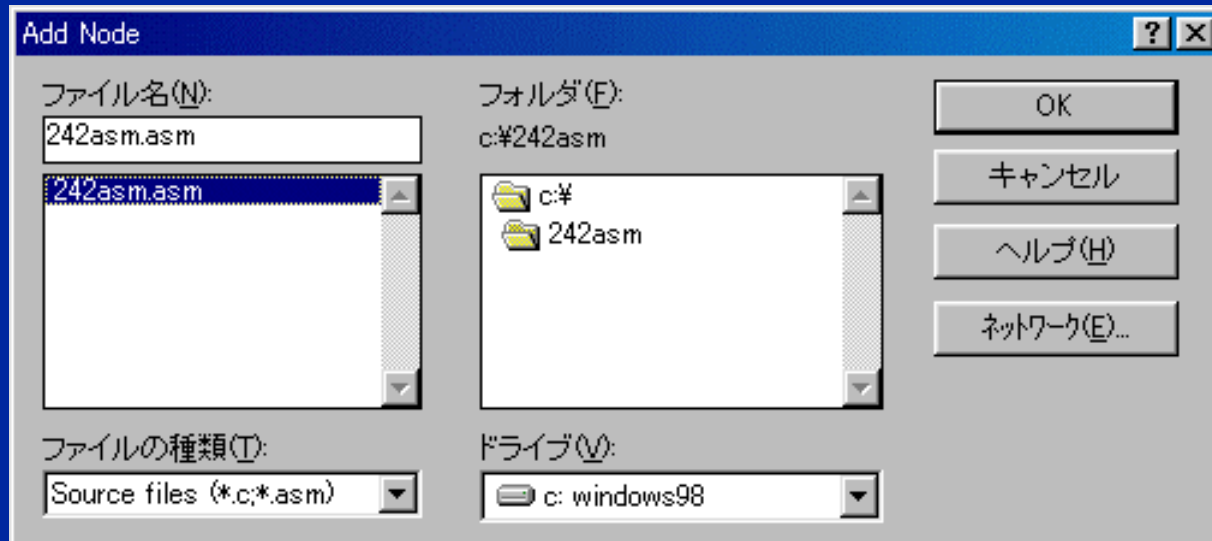
プロジェクトの設定(2)

- Add Nodeボタンをクリックします。



プロジェクトの設定(3)

- 以下の設定にして、OKをクリックします。
 - フォルダ: c:¥242asm
 - ファイル名: 242asm.asm



プロジェクトの設定(4)

- OKボタンをクリックします。

Edit Project

Project

Target Filename
242asm.hex

Include Path

Library Path

Linker Script Path

Development Mode: MPLAB-SIM18C242 Change...

Language Tool Suite: Microchip

Project Files

242asm [.hex]
242asm [.asm]

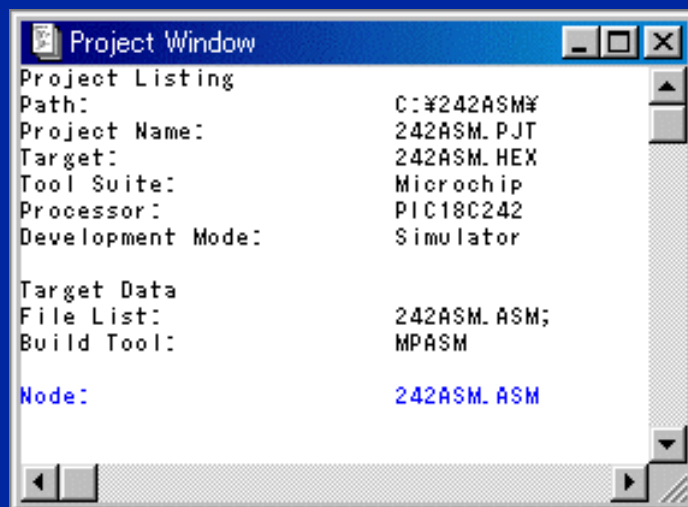
Add Node...
Copy Node...
Delete Node
Build Node
Node Properties...

OK
Cancel
Help



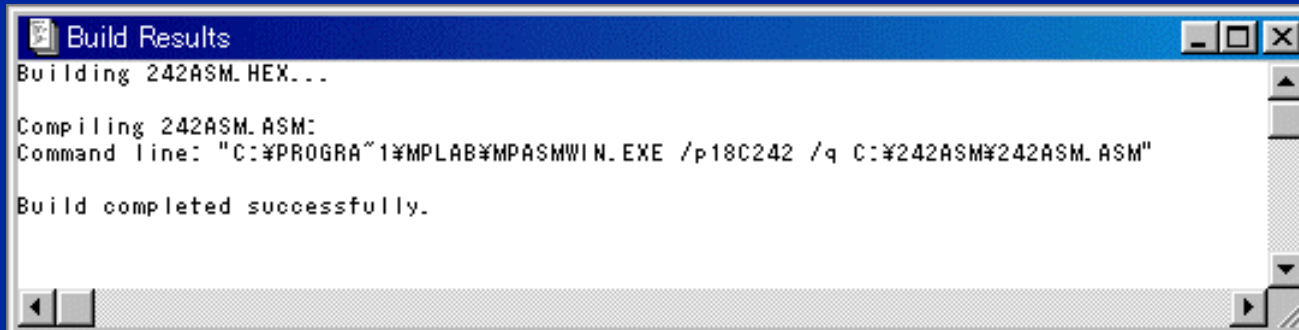
ソース・リストを表示する

- Window→Projectを選択します。
- 242asm.asmをダブル・クリックします。



アセンブル(1)

- Project→Make Projectを選択します。
 - アセンブルがはじまり、その結果が報告されます。
- エラーがないときは、最後の行に Build completed successfully.と表示されます。



```
Build Results
Building 242ASM.HEX...

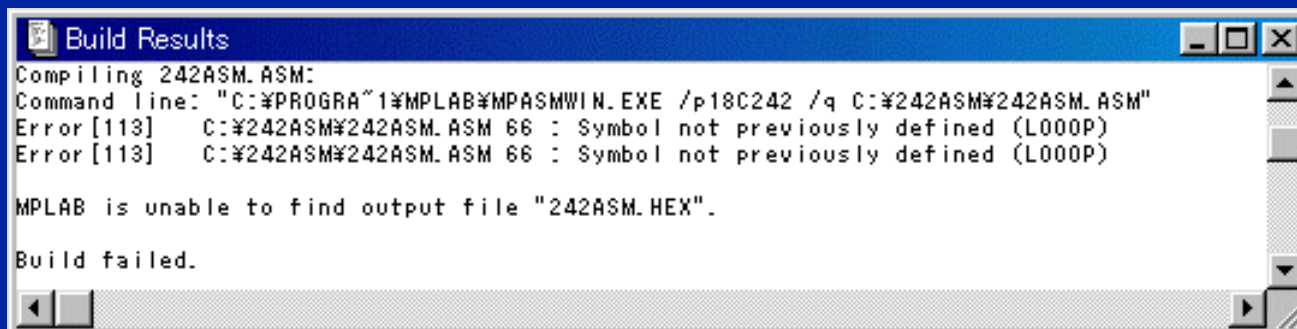
Compiling 242ASM.ASM:
Command line: "C:\PROGRAMS\MPLAB\MPLASMWIN.EXE /p18C242 /q C:\242ASM\242ASM.ASM"

Build completed successfully.
```



アセンブル(2)

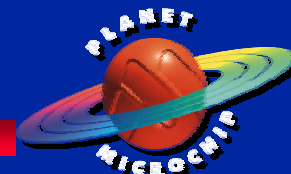
- エラーがあるときはBuild failed.と報告されます。
 - エラー・メッセージの行をダブル・クリックするとソース・リストのエラーのある行にカーソルが跳びます。
 - エラーを修正して、再度コンパイルしましょう。



```
Build Results
Compiling 242ASM.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p18C242 /q C:\242ASM\242ASM.ASM"
Error [113] C:\242ASM\242ASM.ASM 66 : Symbol not previously defined (L000P)
Error [113] C:\242ASM\242ASM.ASM 66 : Symbol not previously defined (L000P)

MPLAB is unable to find output file "242ASM.HEX".

Build failed.
```



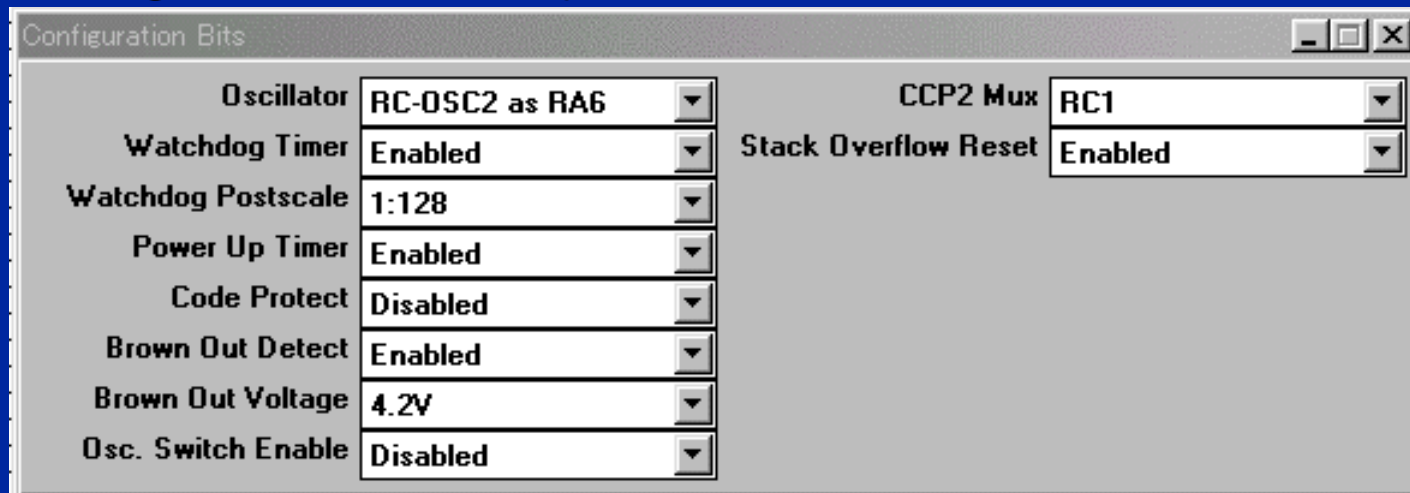
PICSTART Plus(1)

- 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。
 - EIA-574



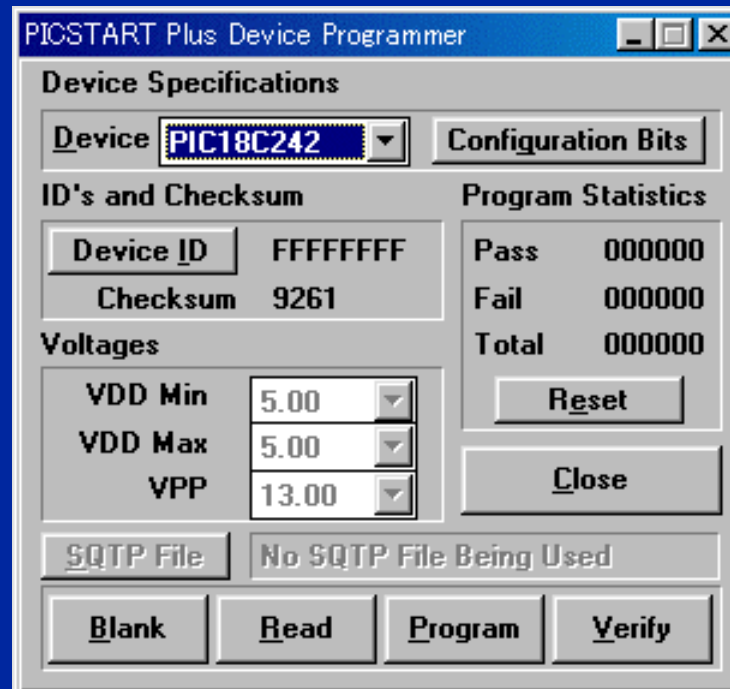
PICSTART Plus(2)

- PICSTART Plus→Enable Programmerを選択します。
- 以下のダイアログが表示されます。
 - PICSTART Plus Device Programmer
 - Configuration Bits
 - Program Memory Window



PICSTART Plus(3)

- デバイスをPICSTART Plusのソケットに挿入します。
- PICSTART Plus Device ProgrammerダイアログのProgramボタンをクリックします。



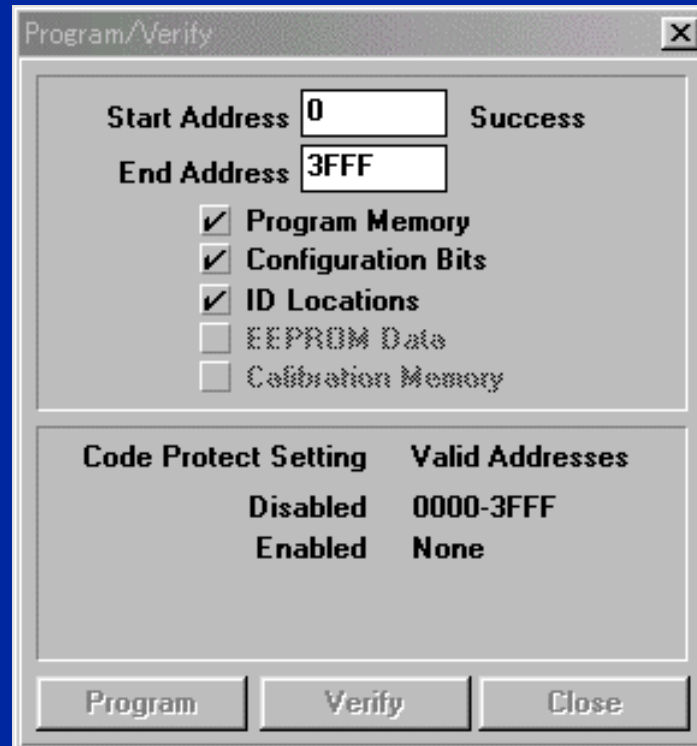
PICSTART Plus(4)

- デバイスがブランクではないときは、下図のように報告されます。
 - 紫外線消去する必要があります。
 - デバイスが正しくソケットに入っているかも確認しましょう。



PICSTART Plus(5)

- エラーなしで書き込みできたときはSuccessと表示されます。



PICSTART Plus(6)

- エラーがあったときは、Failureと報告され、エラー・ログ・ファイルが開きます。

