

แนวทางการพัฒนาโปรแกรม

- การพัฒนาโปรแกรมด้วยภาษาแอสเซมบลี
- การพัฒนาโปรแกรมด้วยภาษา BASIC
- การพัฒนาโปรแกรมด้วยภาษา C

การพัฒนาโปรแกรมด้วยภาษาแอสเซมบลี

```
List    p = 16F877                ; list directive to define processor
#include <p16F877.inc>            ; processor specific variable definitions
    _CONFIG                      ; การกำหนดค่า Configuration
#define output    PORTA,0        ; การกำหนดตัวแปรแทนรีจิสเตอร์
Temp EQU    0x70                ; การกำหนดตัวแปรในหน่วยความจำ
; -----
    ORG    0x000                ; processor reset vector
LABEL    operation    operand    ; comment
        .....
        .....
    END                          ; directive 'end of program'
```

รูปแบบการเขียนค่าคงที่ในแอสเซมบลี

เลขฐานสิบหก (HEX) 0x0A

เลขฐานสิบ (DEC) d'10' หรือ .10

เลขฐานสิบสอง (BIN) b'00001010'

การกำหนดค่า Configuration

`_BODEN_ON` , `_BODEN_OFF`

`_CP_ALL` , `_CP_HALF` , `_CP_UPPER_256` , `_CP_OFF`

`_WRT_ENABLE_ON` , `_WRT_ENABLE_OFF`

Brown-out Reset Enable bit

`_BODEN_ON` `_BODEN_OFF`

Flash Program Memory Code Protection bits

`_CP_ALL` `_CP_HALF` `_CP_UPPER_256` `_CP_OFF`

Flash Program Memory Write Enable bit

`_WRT_ENABLE_ON` , `_WRT_ENABLE_OFF`

Power-up Timer Enable bit

`_PWRTTE_OFF` `_PWRTTE_ON`

Watchdog Timer Enable bit

`_WDT_ON` `_WDT_OFF`

Oscillator Selection bits

`_LP_OSC` `_XT_OSC` `_HS_OSC` `_RC_OSC`

Background Debugger Mode

`_DEBUG_ON` `_DEBUG_OFF`

Data EE Memory Code Protection

`_CPD_ON` `_CPD_OFF` `_LVP_ON` `_LVP_OFF`

REGISTER 12-1: CONFIGURATION WORD (ADDRESS 2007h)⁽¹⁾

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
bit13										bit0			
bit 13-12, bit 5-4		CP1:CP0: FLASH Program Memory Code Protection bits ⁽²⁾ 11 = Code protection off 10 = 1F00h to 1FFFh code protected (PIC16F877, 876) 10 = 0F00h to 0FFFh code protected (PIC16F874, 873) 01 = 1000h to 1FFFh code protected (PIC16F877, 876) 01 = 0800h to 0FFFh code protected (PIC16F874, 873) 00 = 0000h to 1FFFh code protected (PIC16F877, 876) 00 = 0000h to 0FFFh code protected (PIC16F874, 873)											
bit 11		DEBUG: In-Circuit Debugger Mode 1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins 0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger.											
bit 10		Unimplemented: Read as '1'											
bit 9		WRT: FLASH Program Memory Write Enable 1 = Unprotected program memory may be written to by EECON control 0 = Unprotected program memory may not be written to by EECON control											
bit 8		CPD: Data EE Memory Code Protection 1 = Code protection off 0 = Data EEPROM memory code protected											
bit 7		LVP: Low Voltage In-Circuit Serial Programming Enable bit 1 = RB3/PGM pin has PGM function, low voltage programming enabled 0 = RB3 is digital I/O, HV on MCLR must be used for programming											
bit 6		BODEN: Brown-out Reset Enable bit ⁽³⁾ 1 = BOR enabled 0 = BOR disabled											
bit 3		PWRTE: Power-up Timer Enable bit ⁽³⁾ 1 = PWRT disabled 0 = PWRT enabled											
bit 2		WDTE: Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled											
bit 1-0		FOSC1:FOSC0: Oscillator Selection bits 11 = RC oscillator 10 = HS oscillator 01 = XT oscillator 00 = LP oscillator											

Note 1: The erased (unprogrammed) value of the configuration word is 3FFFh.

Note 2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

Note 3: Enabling Brown-out Reset automatically enables Power-up Timer (PWRTE), regardless of the value of bit PWRTE. Ensure the Power-up Timer is enabled any time Brown-out Reset is enabled.

Source File : ASM_EXAM\232.ASM

Processor Type : None

Error File : Yes ASM_EXAM\232.ERR

Cross Reference File : No

Listing File : Yes ASM_EXAM\232.LST

Hex Dump Type : INHX8M ASM_EXAM\232.HEX

Assemble to Object File : No

,Tab : Move Cursor
F1 : Help

Esc : Quit
F10 : Assemble

Type the name of your source file.

Source File Name:

D:\ASM\232.ASM

Browse...



MICROCHIP

Options:

Radix:

- Default
- Hexadecimal
- Decimal
- Octal

Warning Level:

- Default
- All Messages
- Warnings and Errors
- Errors Only

Hex Output

- Default
- INHX8M
- INHX8S
- INHX32

Generated Files:

- Error File
- List File
- Cross Reference File
- Object File

Case Sensitive

Macro Expansion:

- Default
- On
- Off

Processor: Default

Tab Size:

8

Extra Options:

Empty text input field for extra options.

Exit

Assemble

Save Settings on Exit

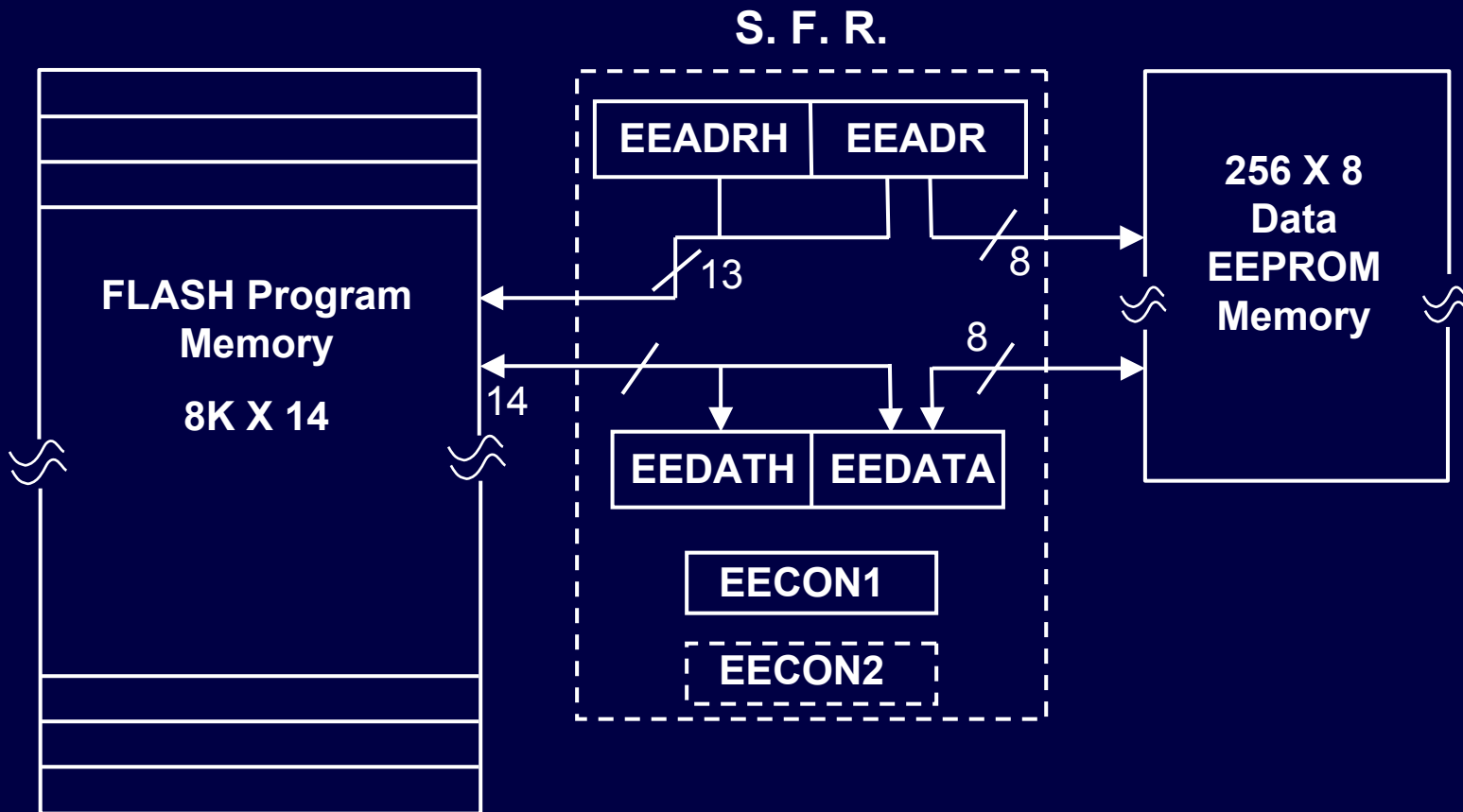
Help

การใช้งานพอร์ต อินพุต/เอาต์พุต

การกำหนดการทำงานของ I/O Ports

```
MOVLW 0xCF      ; ค่าสำหรับกำหนดการทำงานของ PortA  
                ; 11001111  
MOVWF TRISA     ; Set RA<3:0> as inputs  
                ; RA<5:4> as outputs  
                ; RA<7:6> as inputs
```

FLASH Program and Data EEPROM Memory



Data EEPROM Memory: Reading Locations

- Program sequence to read Internal Data EEPROM:
 - Write the desired address to EEADR.
 - Clear the EEPGD bit, EECON1<7>, which will select Data EEPROM.
 - Set the RD bit, EECON1<0>.
 - Data will be available in the EEDATA register in the next instruction cycle.

Data EEPROM Memory: Reading Locations

```
Program:
BSF      STATUS , RP1      ;Move to Bank2
BCF      STATUS , RP0
MOVLW   ADDRESS          ;Write data memory address
MOVWF   EEADR             into the EEADR register
BSF      STATUS , RP0      ;Move to Bank3
BCF      EECON1 , EEPGD    ;Point to Data Memory
BSF      EECON1 , RD       ;Start the Read Operation,
                           data will be available in
                           the very next Tcy
BCF      STATUS , RP0
MOVF    EEDATA , W        ;Move the data into W
```

Data EEPROM Memory: Writing to Locations

● Sequence to write data to Internal Data EEPROM:

- Write the desired address to EEADR.
- Write the desired data to EEDATA.
- Clear the EEPGD bit, EECON1<7> which will select the data memory access.
- Set the WREN bit, EECON1<2>.
- Disable all interrupts.
- Write 55h followed by AAh to EECON2.
- Set the WR bit, EECON1<1>.
- Clear the WREN bit, EECON1<2>.
- Interrupts can now be enabled.
- Wait for WR bit to clear or EEIF to set, which indicates write operation has completed.

Data EEPROM Memory: Writing to Locations

```
BSF    STATUS ,RP1           ;Move to Bank2
BCF    STATUS ,RP0
MOVLW  ADDRESS              ;Write desired address to EEADR
MOVWF  EEADR                ;Write desired data to EEDATA
MOVLW  VALUE
MOVWF  EEDATA              ;Move to Bank3
BSF    STATUS ,RP0          ;Enable EEPROM writes
BSF    EECON1 ,WREN        ;Disable interrupts
BCF    EECON1 ,EEPGD       ;Point to data memory
BCF    INTCON ,GIE
MOVLW  55h                  ;Next five instructions are required
MOVWF  EECON2              ;sequence to initiate a write sequence
MOVLW  AAh
MOVWF  EECON2              ;Initiate the write sequence
BSF    EECON1 ,WR          ;Enable interrupts
BSF    INTCON ,GIE        ;Disable EEPROM writes, does not affect
BCF    EECON1 ,WREN        the current write cycle
```

FLASH Program Memory: Reading Locations

- Write the 8 LSbs of the desired address to EEADR.
- Write the 5 MSbs of the desired address to EEADRH.
- Set the EEPGD bit, EECON1<7> to access Program FLASH Memory.
- Set the RD bit, EECON1<0>.
- Next two instruction must be set as NOPs.
- Data will be available in the EEDATH:EEDATA registers in the instruction after the NOPs.

FLASH Program Memory: Reading Locations

```
BSF     STATUS , RP1           ;Move to Bank2
BCF     STATUS , RP0
MOVLW  ADDRH                   ;Write data memory address into
MOVWF  EEADRH                   ;the EEADRH:EEADR register
MOVLW  ADDRL
MOVWF  EEADR
BSF     STATUS , RP0           ;Point to program memory
BSF     EECON1 , EEPGD
BSF     EECON1 , RD            ;Start the Read Operation, data will
NOP                                           ;be available in the third TCY
NOP
BCF     STATUS , RP0
MOVF   EEDATA , W
MOVWF  Temp
MOVF   EEDATH , W
```


FLASH Program Memory: Writing to Locations

- Write the desired address to EEADRH:EEADRL.
- Write the desired data to EEDATH:EEDATA.
- Set the EEPGD bit, EECON1<7> to access Program FLASH Memory.
- Set the WREN bit, EECON1<2>.
- Disable all interrupts.
- Write 55h to EECON2 followed by AAh to EECON2.
- Set the WR bit, EECON1<1>.
- Next two instructions must be NOPs.
- CPU now halts while memory is programmed (~2mS), this is NOT sleep mode as the clocks and peripherals continue to run.
- When the write is completed, program execution starts at the instruction just after the two NOPs (above).
- Clear the WREN bit, EECON1<2>.

FLASH Program Memory: Writing to Locations

```
BANKSEL    EEADDRH           ;Move to Bank2
MOVLW     ADDRH             ;Write desired address to EEADDRH:EEADR
MOVWF     EEADDRH
MOVLW     ADDRL             ;Write desired data to EEDATH:EEDATA
MOVWF     EEADR
MOVLW     DATAH
MOVWF     EEDATH
MOVLW     DATAL
MOVWF     EEDATA
BSF       STATUS,RP0       ;Move to Bank3
BSF       EECON1,WREN      ;Enable EEPROM writes
BSF       EECON1,EEPGD     ;Point to program memory
BCF       INTCON,GIE      ;Disable interrupts
MOVLW     55h              ;Next five instructions are required
MOVWF     EECON2           ;sequence to initiate a write sequence
MOVLW     AAh
MOVWF     EECON2
BSF       EECON1,WR       ;Initiate the write sequence
NOP       ;Instruction is executed normally
NOP       ;Instruction is ignored, processor halts
BSF       INTCON,GIE      ;Enable interrupts
BCF       EECON1,WREN     ;Disable EEPROM writes
```

การคำนวณหาค่า Baud Rate

SYNC	BREG=0 (Low Speed)	BREG=1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	(Asynchronous) Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

X = value in SPBRG (0 to 255)

BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	FOSC = 4 MHz			FOSC = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	FOSC = 4 MHz			FOSC = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

การพัฒนาโปรแกรมด้วยภาษา Basic (Pic Basic Pro)

- การกำหนดตัวแปร

Label var Size

เช่น

Temp var word ; ตัวแปรประเภท word

Temp1 var byte ; ตัวแปรประเภท byte

Temp2 var bit ; ตัวแปรประเภท bit

SW_1 var PORTA.0 ; ตัวแปรที่เป็นรีจิสเตอร์

- การกำหนดตัวแปรชนิด Arrays

Label var Size[Number of elements]

เช่น **Cat var byte[10]**

Dog var bit[8]

- การกำหนดค่าคงที่ Constants

Label CON Constant expression

เช่น dataA CON 8

- การกำหนดตัวแปร String จะต้องเขียนอยู่ในเครื่องหมาย “ ”

เช่น “Hello”

- การกำหนด Labels ซึ่งในการเขียนโปรแกรมที่มีการกระโดดจำเป็นต้องมี Labels เพื่อบอกตำแหน่งปลายทาง รูปแบบคือ ชื่อลาเบล ตามด้วยเครื่องหมาย colon(:) เช่น

here: serout PORTC.6,T9600,[“Hello”,13,10]

goto here

- การเขียน Comments จะใส่เครื่องหมาย single quote(‘) หรือ semi-colon (;) นำหน้าข้อความที่ไม่ต้องการให้ CPU ประมวลผล

เช่น Lcdout “Hello” ‘ Output String “Hello”

Lcdout “Hello” ; Output String “Hello”

- การเขียนค่าคงที่เลขฐาน สอง,สิบ และ สิบหก

ฐานสองจะใช้เครื่องหมาย % นำหน้าตัวเลข เช่น **%10010100**

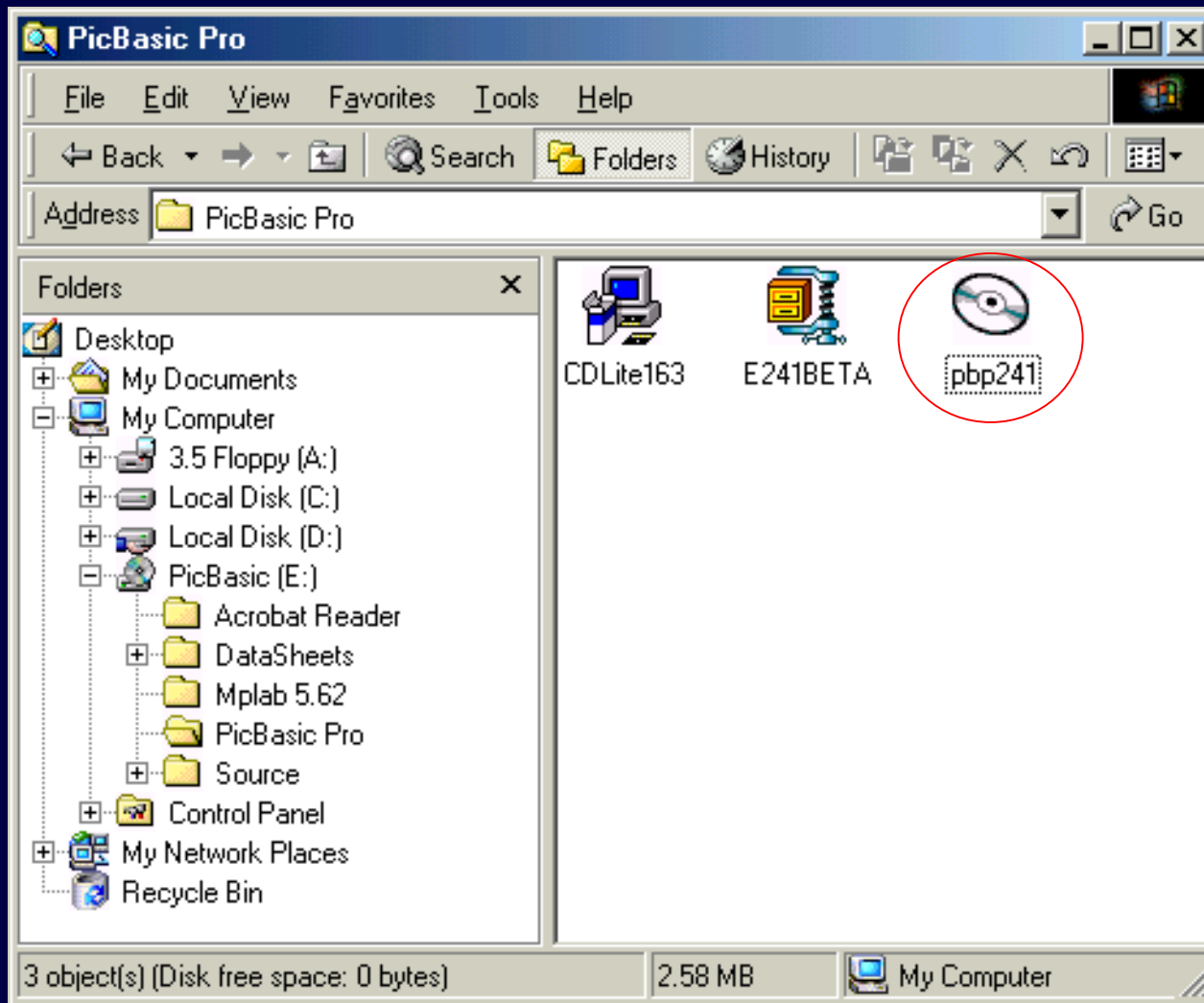
ฐานสิบจะเขียนตามปกติ เช่น **100**

ฐานสิบหกจะใช้เครื่องหมาย \$ นำหน้าตัวเลข เช่น **\$9F**

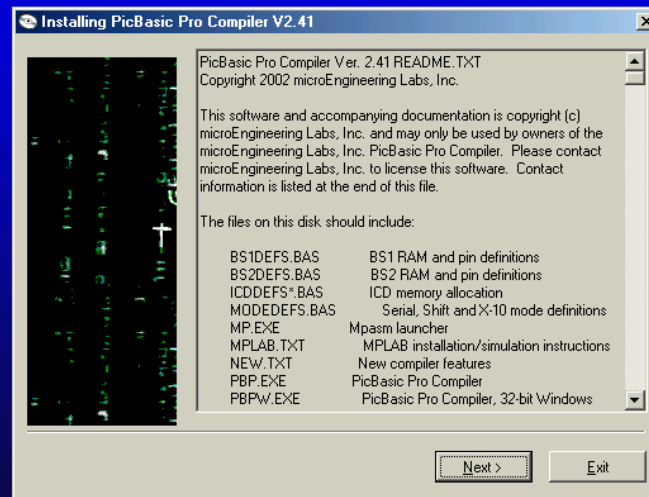
- การกำหนด INCLUDE จะต้องใส่ไว้บนสุดของโปรแกรม

เช่น **INCLUDE "modedefs.bas"**

การติดตั้งโปรแกรม PicBasic Pro compile V2.41



PicBasic Pro Compile V2.41



Installing PicBasic Pro Compiler V2.41



PicBasic Pro Compile V2.41

Destination Directory

C:\Pbp241

Required: 5895 K

Browse...

< Prev

Start

Exit