

Miscela di assembler e BASIC (I)

Arrivati a questo punto, conosciamo tutte le istruzioni e le strutture dei compilatori BASIC per PIC e del linguaggio assembler MPASM, quindi siamo in condizione di decidere quale preferire. Come avrete potuto verificare, nel BASIC alcune operazioni complesse sono molto facilitate, per contro non è possibile fare ciò che non è previsto nel linguaggio stesso. L'assembler invece, con il suo ridotto e semplice insieme di istruzioni, permette al programmatore di gestire tutti i dispositivi del PIC senza restrizioni, però è necessaria una conoscenza approfondita di quest'ultimo, non solo per sfruttarli al meglio, ma anche per farli funzionare.

La soluzione consiste quindi nell'utilizzare in ogni momento il sistema più adeguato; questo è possibile perché il PicBasicPlus, così come il LetPicBasic, permette di utilizzare le istruzioni assembler all'interno del codice BASIC. Inizieremo nel presente capitolo a mostrarvi degli esempi, e in un capitolo successivo parleremo delle agevolazioni che derivano dal miscelare l'assembler con il BASIC.

Modo di inserire istruzioni assembler

Come nel LetPicBasic, anche nel PicBasicPlus esiste un modo

```

DIM A,B as Byte
B=328
A=dig B,2           ; Su A rimane il valore della seconda posizione: 3
ASH                ; Da questo punto iniziano le istruzioni assembler
  nop
  nop
  nop
ENDASH
    
```

PICBASIC PLUS COMPILED OK. 58 Words used
28 Variables used in the DEFAULT 16F84 from a possible 68

Modo di inserire istruzioni assembler nel PicBasicPlus.

```

DIM A,B as Byte
B=328
A=dig B,2           ; Su A rimane il valore della seconda posizione: 3
@ nop              ; Ogni istruzione assembler deve essere preceduta
@ nop              ; dal carattere "@"
@ nop
    
```

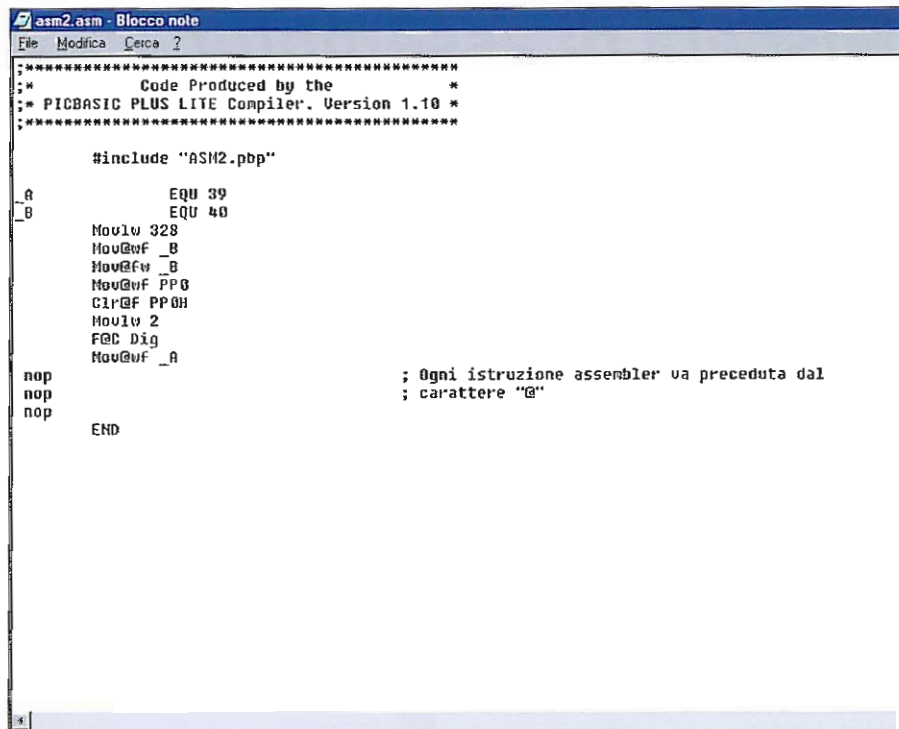
PICBASIC PLUS COMPILED OK. 58 Words used
28 Variables used in the DEFAULT 16F84 from a possible 68

Secondo modo di inserire istruzioni assembler nel PicBasicPlus.

rapido e semplice per inserire istruzioni assembler. Si tratta di utilizzare le parole chiave ASM e ENDASM, tra le quali si scrivono le istruzioni assembler desiderate. Lo stesso effetto si ottiene se davanti a ogni istruzione assembler, si scrive il carattere "@". Come potrete notare in entrambi i casi è stato scelto un programma già visto in un capitolo precedente e sono state aggiunte tre istruzioni "nop" che non modificano altro che il tempo di esecuzione. Le istruzioni passeranno direttamente all'assembler senza l'intervento del compilatore del BASIC. In questo modo se scriviamo male un'istruzione, l'editor BASIC ci avviserà cambiando il colore delle istruzioni visualizzate sul monitor, però il compilatore non produrrà errori. Notate inoltre che il compilatore accetta anche il modo tipico dell'assembler di inserire i commenti, quindi sarà più semplice non sbagliare.

Istruzioni assembler trattate come BASIC

Sino ad ora la miscela di assembler e BASIC è stata fatta in modo, per così dire, puro; possiamo fare di più arrivando a mescolare istruzioni e strutture sulla stessa linea di codice. Ad esempio è stata inserita una istruzione assembler



```
asm2.asm - Blocco note
File Modifica Cerca ?
*****
;* Code Produced by the
;* PICBASIC PLUS LITE Compiler, Version 1.10 *
*****

#include "ASM2.pbp"

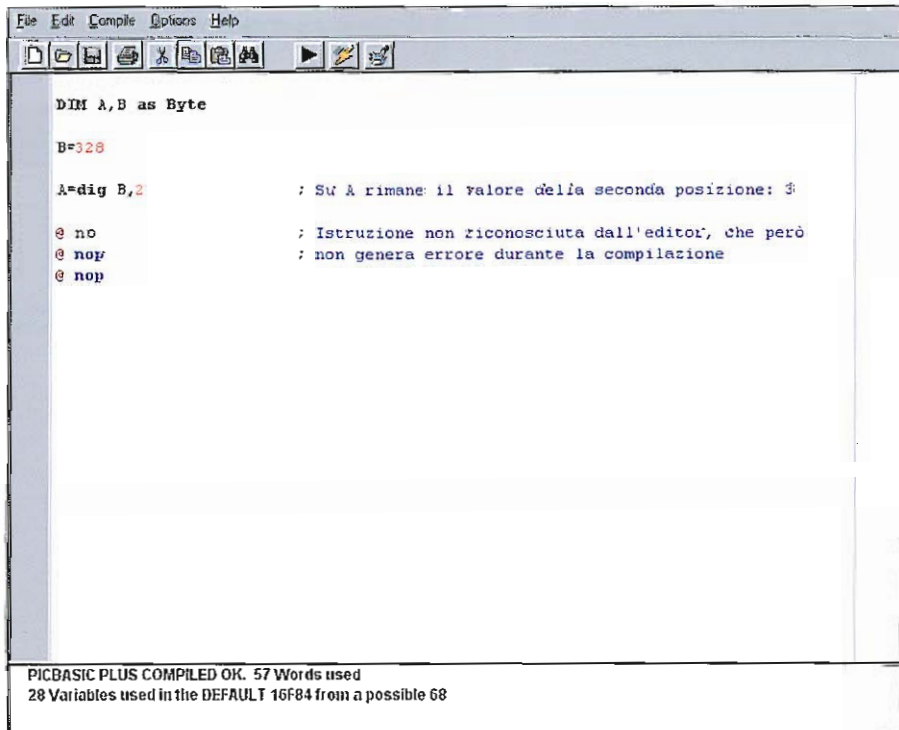
_A EQU 39
_B EQU 40

Moulw 328
Mouwf _B
Mouwf _B
Mouwf PP0
Circf PP0H
Moulv 2
FDC Dig
Mouwf _A

nop ; Ogni istruzione assembler va preceduta dal
nop ; carattere "@"
nop

END
```

Le istruzioni assembler non sono "tradotte" dal compilatore.



```
File Edit Compile Options Help
D O P X R A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` { | } ~

DIM A,B as Byte

B=328

A=dig B,2 ; Su A rimane il valore della seconda posizione: 3

@ no ; Istruzione non riconosciuta dall'editor, che però
@ nop ; non genera errore durante la compilazione
@ nop

PICBASIC PLUS COMPILED OK. 57 Words used
28 Variables used in the DEFAULT 16F84 from a possible 68
```

Compilando un'istruzione assembler scritta male, non si produrrà errore.