

Istruzioni di ingresso e uscita dei dati

Il LetPicBasicLite utilizza delle direttive e dei comandi grazie ai quali, non bisogna preoccuparsi degli aspetti interni del microcontroller.

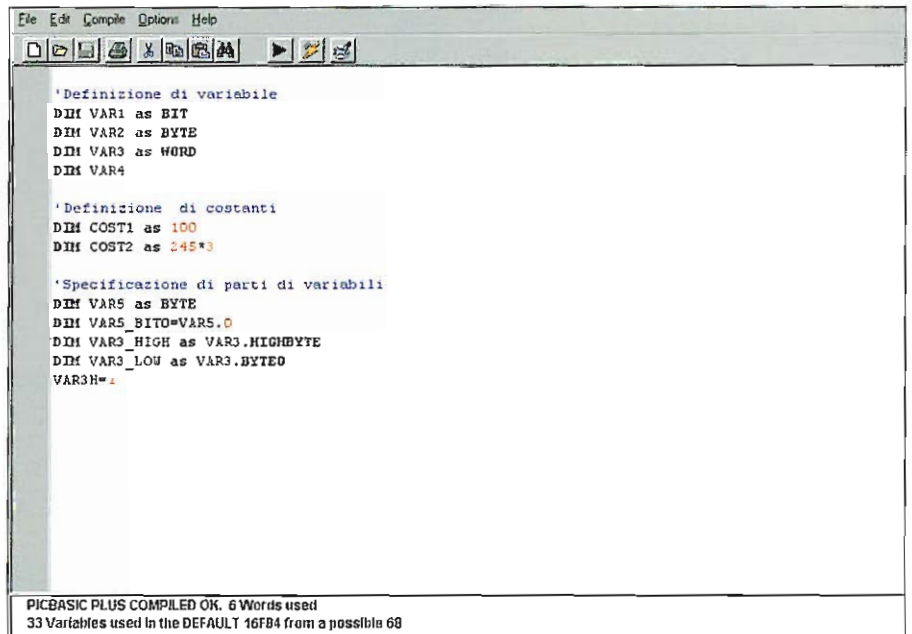
In assembler, invece, è necessario conoscere come sono organizzati sia il percorso dei dati sia quello delle istruzioni, per capire il modo di procedere.

Il LetPicBasicPlus fornisce le peculiarità di un linguaggio come il BASIC per la gestione dei dati e delle strutture di controllo, però non dimentica del tutto che stiamo lavorando con i microcontroller.

Quindi, in questo capitolo, tutti i concetti vi sembreranno familiari e già visti, dato che il LetPicBasicPlus raccoglie una miscela del LetPicBasicLite e dell'assembler, per cui non sarà difficile utilizzarlo se conoscete quali sono le cose che cambiano.

Dichiarazione di variabili

Prima di utilizzare una qualsiasi variabile è necessario dichiararla. In assembler si utilizza la direttiva EQU, in cui bisogna specificare la posizione della memoria dove verrà posizionata la variabile; nel LetPicBasicLite invece si utilizza DIM. Anche nel LetPicBasicPlus si utilizza DIM, però le sue possibilità sono maggiori. Una delle opzioni è dichiarare



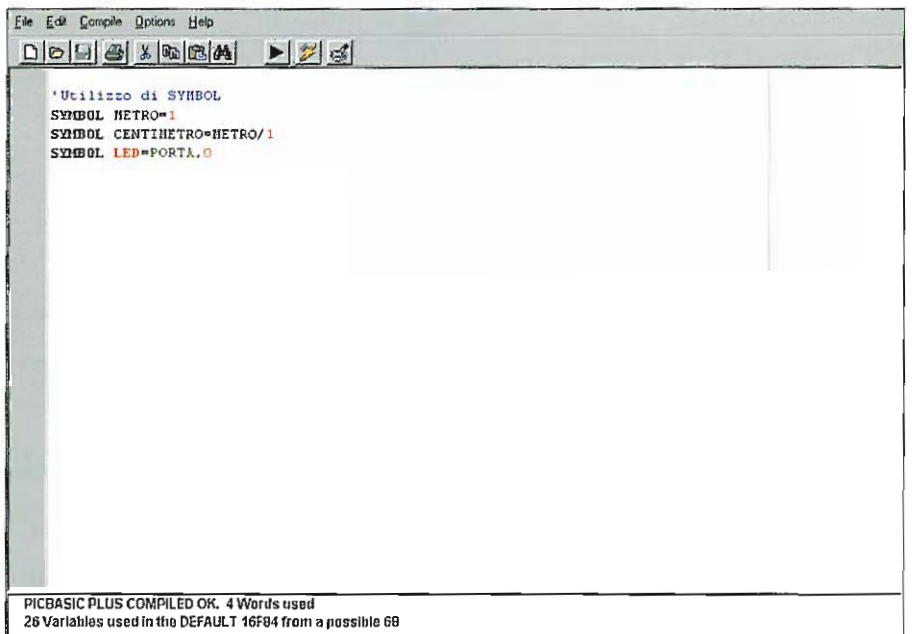
```
File Edit Compile Options Help
[Icons]
'Definizione di variabile
DIM VAR1 as BIT
DIM VAR2 as BYTE
DIM VAR3 as WORD
DIM VAR4

'Definizione di costanti
DIM COST1 as 100
DIM COST2 as 245*3

'Specificazione di parti di variabili
DIM VAR5 as BYTE
DIM VAR5_BIT0=VAR5.0
DIM VAR5_HIGH as VAR5.HIGHBYTE
DIM VAR5_LOW as VAR5.BYTE0
VAR5H=
```

PICBASIC PLUS COMPILED OK. 6 Words used
33 Variables used in the DEFAULT 16F84 from a possible 68

Dichiarazione di variabili nel LetPicBasicPlus.



```
File Edit Compile Options Help
[Icons]
*Utilizzo di SYMBOL
SYMBOL METRO=1
SYMBOL CENTIMETRO=METRO/1
SYMBOL LED=PORTA.0
```

PICBASIC PLUS COMPILED OK. 4 Words used
26 Variables used in the DEFAULT 16F84 from a possible 68

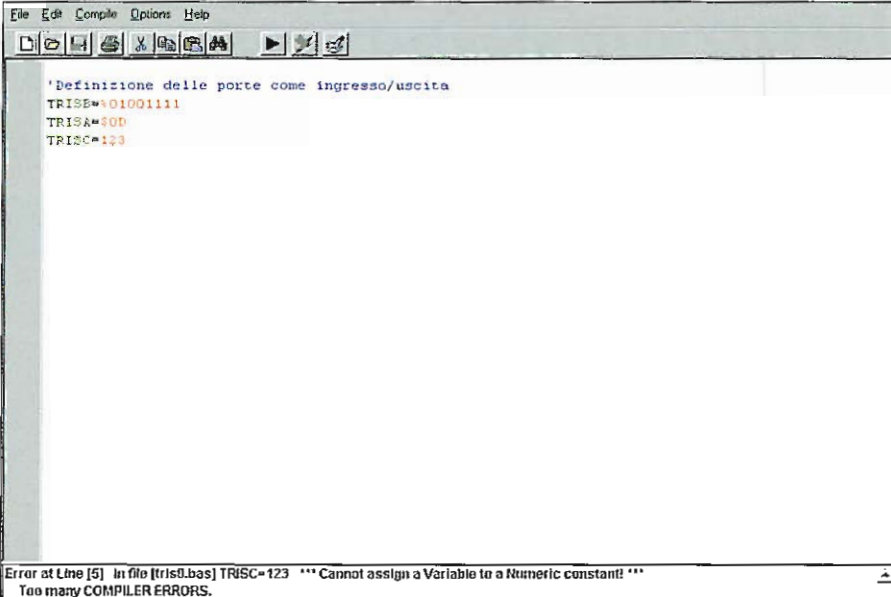
Il comando SYMBOL.

solamente la dimensione che avrà questa variabile, che può essere BIT, BYTE o WORD. Se non si specifica la dimensione, si prende il BYTE per default. Il nome delle variabili ha una dimensione massima di 32 caratteri e può contenere il tratto basso (_) e i numeri, anche se non può iniziare con questi ultimi. Inoltre, le maiuscole non si differenziano dalle minuscole.

Se al posto di una dimensione si specifica un valore o un'espressione, ciò che viene definito non è una variabile ma una costante, per cui non occupa spazio nella memoria.

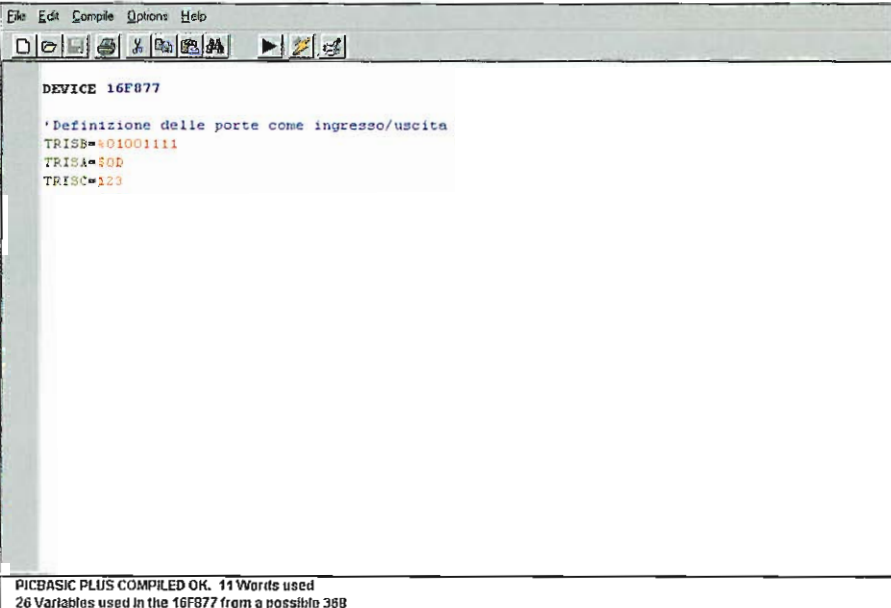
È possibile inoltre fare una combinazione di entrambi e specificare, ad esempio, un bit all'interno di una variabile, per lavorare più comodamente. La variabile occuperà uno spazio, però la definizione del bit non sarà uno spazio in più, ma semplicemente un "alias".

All'interno di questo alias ve ne sono alcuni già tipizzati, quali HIGHBYTE, BYTE1, LOWBYTE e BYTE0. I primi due fanno riferimento al byte più significativo di una variabile di dimensione WORD, e i secondi due al byte meno significativo. Se conosciamo anche come il compilatore traduce la nostra istruzione BASIC in assembler, possiamo semplificarla ancora di più, dato che per accedere alla parte alta di una variabile di dimensione WORD basta aggiungere al nome una "H": in questo modo il compilatore le passa in assembler. Le variabili verranno



```
File Edit Compile Options Help
'Definizione delle porte come ingresso/uscita
TRISB=01001111
TRISA=30D
TRISC=123
Error at Line [5] in file [tris0.bas] TRISC=123 *** Cannot assign a Variable to a Numeric constant! ***
Too many COMPILER ERRORS.
```

Per default il PIC è il 16F84.



```
File Edit Compile Options Help
DEVICE 16F877
'Definizione delle porte come ingresso/uscita
TRISB=01001111
TRISA=30D
TRISC=123
PICBASIC PLUS COMPILED OK. 11 Words used
26 Variables used in the 16F877 from a possible 368
```

Definizione delle porte di ingresso/uscita.

posizionate nella memoria dei dati secondo la loro dichiarazione nel programma BASIC, in modo che, se il microcontroller ha più di un banco primario, ne riempirà

uno prima di passare all'altro. Nell'assembler eravamo noi che dovevamo cambiare banco al momento opportuno, nel LetPicBasicPlus questo avviene in modo automatico. L'unico