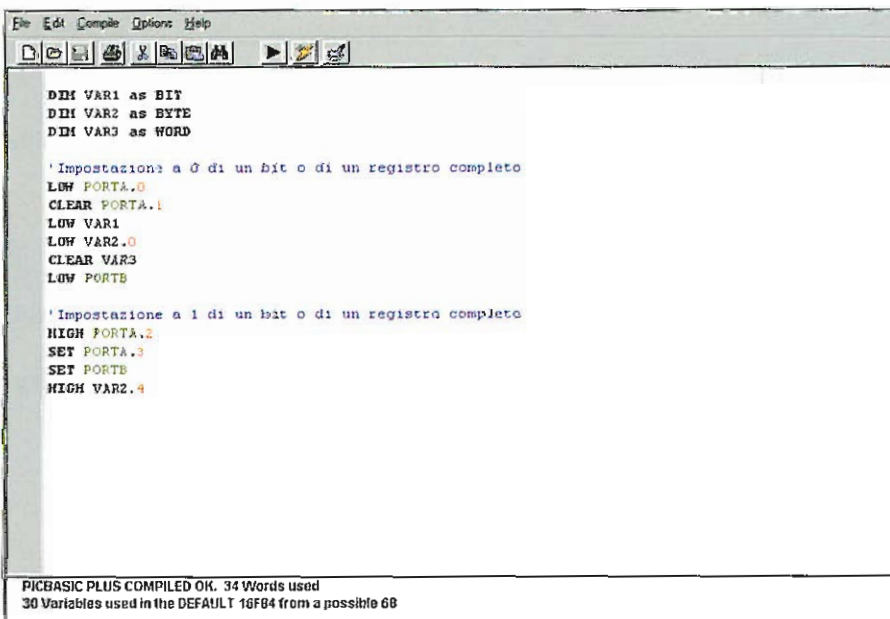


```
File Edit Compile Options Help
[Icons]
'Definizione di linee indipendenti come ingresso e uscita
OUTPUT PORTA.0
OUTPUT PORTA.1
INPUT PORTE.3

TRISB.4=0
TRISB.5=1

PICBASIC PLUS COMPILED OK. 34 Words used
30 Variables used in the DEFAULT 16F84 from a possible 68
```

Definizione delle linee indipendenti delle porte.



```
File Edit Compile Options Help
[Icons]
DIM VAR1 as BIT
DIM VAR2 as BYTE
DIM VAR3 as WORD

'Impostazione a 0 di un bit o di un registro completo
LOW PORTA.0
CLEAR PORTA.1
LOW VAR1
LOW VAR2.0
CLEAR VAR3
LOW PORTE

'Impostazione a 1 di un bit o di un registro completo
HIGH PORTA.2
SET PORTA.3
SET PORTE
HIGH VAR2.4

PICBASIC PLUS COMPILED OK. 34 Words used
30 Variables used in the DEFAULT 16F84 from a possible 68
```

Istruzioni di impostazione a 0 e a 1.

problema si potrebbe verificare se una variabile di dimensione WORD si posizionasse fra due banchi, ma in questo caso saremmo avvisati al momento della compilazione

e troveremmo facilmente una soluzione. Il comando SYMBOL si utilizzerà come nel LetPicBasicLite, essendo questo un altro modo di assegnare nomi alle costanti.

Definizione delle porte

Il comando DEFINE non esiste nel LetPicBasicPlus, dove la definizione delle porte come ingresso o uscita si fa in un altro modo. Esistono due modi per realizzare queste definizioni. Uno di questi è simile alla procedura dell'assembler, in cui si tiene conto del nome dei registri di configurazione delle porte. Quindi, potremo assegnare valori a TRISB per configurare la Porta B, e come in assembler, il valore 0 configurerà le linee come uscita e il valore 1 come ingresso.

La forma di rappresentazione numerica continuerà a essere in binario (%), decimale o esadecimale (\$). Ricordiamo che se non si specifica il comando DEVICE, il microcontroller per default utilizzerà il PIC16F84, in cui alcuni dispositivi come la Porta C non esistono, e questo genererà errore. Invece di modificare il carattere delle porte al completo, ci potrebbe tornare utile in un particolare momento del programma cambiare solamente il modo di attuazione di una delle linee. Per fare questo, disponiamo di due nuove istruzioni: INPUT e OUTPUT. Con la prima, una linea di una porta si configura come ingresso e con la seconda, questa linea si configura come uscita. Tuttavia, questa rimane solamente una configurazione: definiamo una linea come uscita però non specifichiamo che valore deve avere, né acquisiamo il valore nel caso di averla definita come ingresso. La definizione di bit indipendenti

Assegnazione di valori alle linee di una porta

Le istruzioni sono le stesse del LetPicBasicLite: LOW o CLEAR per impostare a 0, e HIGH o SET per impostare a 1. Mentre in precedenza si potevano utilizzare solo per assegnare valori a linee indipendenti, in questo caso è possibile attuare su variabili complete, portandole a 0 oppure a 1 rispettivamente. Per utilizzare queste istruzioni non è necessario che la variabile sia una porta.

Quando si utilizzano con le porte, le linee di queste sono automaticamente configurate come ingresso o uscita, a seconda dell'istruzione. Nello stesso programma si può cambiare il valore e la configurazione di un registro tutte le volte che lo riteniamo necessario. Al posto delle istruzioni che agiscono su bit indipendenti, qui è possibile assegnare valori alla porta al completo, dopo averla precedentemente configurata, però bisogna tener conto che se tutti i bit della porta non sono di uscita, dobbiamo controllare noi stessi i valori inseriti, perché il compilatore non ci avviserà dell'errore. La procedura corretta prevede l'utilizzo di maschere per eliminare la possibilità di danneggiare il microcontroller. Se una porta è stata configurata come ingresso, avrà il valore che arriva dalle periferiche, per acquisirne il valore è sufficiente assegnarlo a un registro per poi trattarlo di conseguenza o verificarlo direttamente tramite il valore di questo registro, ad esempio, con una istruzione IF. Nell'esempio, i due modi di verificare il dato della Porta B sono identici.

```
File Edit Compile Options Help
[Icons]
DEVICE 16F877

' Carico di un valore su di una porta
TRISB=#11110000
PORTB=#33

TRISC=0
PORTC=#3F

PICBASIC PLUS COMPILED OK. 15 Words used
26 Variables used in the 16F877 from a possible 368
```

Durante la configurazione bisogna fare attenzione a lavorare con tutti i bit delle porte insieme.

```
File Edit Compile Options Help
[Icons]
DIM VAR1 as BYTE
DIM VAR2 as BIT

' Configurazione di una porta come ingresso
TRISE=#FF

' Acquisizione dei dati della porta
VAR1=PORTE

' Due modi di fare la stessa cosa
IF VAR1=#34 THEN VAR2=1
ELSE VAR2=0

IF PORTE=#34 THEN VAR2=1
ELSE VAR2=0

PICBASIC PLUS COMPILED OK. 16 Words used
28 Variables used in the DEFAULT 16F84 from a possible 60
```

Acquisizione dei dati da una porta di ingresso.

si può realizzare anche specificando il bit del registro TRIS. Esistono due differenze per quanto riguarda il farlo in un modo oppure in un altro: INPUT e OUTPUT danno maggiore

chiarezza a ciò che si sta facendo, perché non ci si deve preoccupare se gli 0 significano ingresso o uscita, così come per gli 1. Tuttavia, produce più codice che l'utilizzo di TRIS.