Semplici acquisizioni di dati

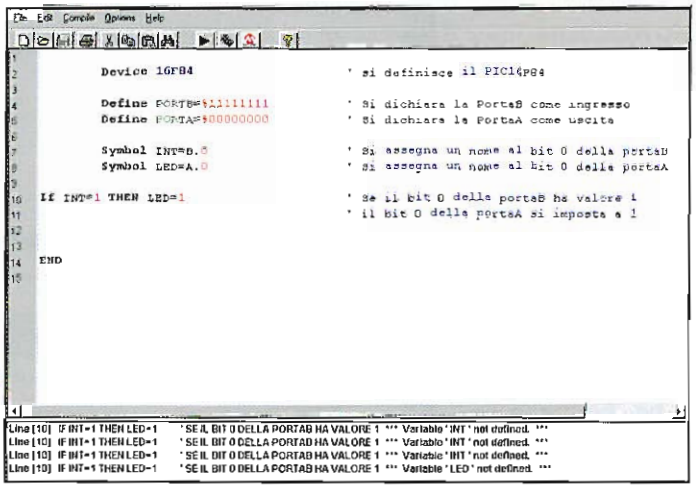
In precedenza abbiamo dedicato un capitolo all'acquisizione dei dati dalle porte di ingresso/uscita. Praticamente questa è l'unica via di comunicazione che abbiamo con l'esterno. Immaginate ora che ci interessi solamente il valore di un determinato bit della porta, oppure che il programma non possa continuare fino a quando non arriva un valore tramite un pin specifico, oppure che al posto di un singolo pulsante o interruttore dobbiamo collegare una tastiera.

Con quanto abbiamo visto sinora potremmo già realizzare il programma, però il LetPicBasic ci fornisce delle istruzioni specifiche per questi casi, che vedremo di seguito.

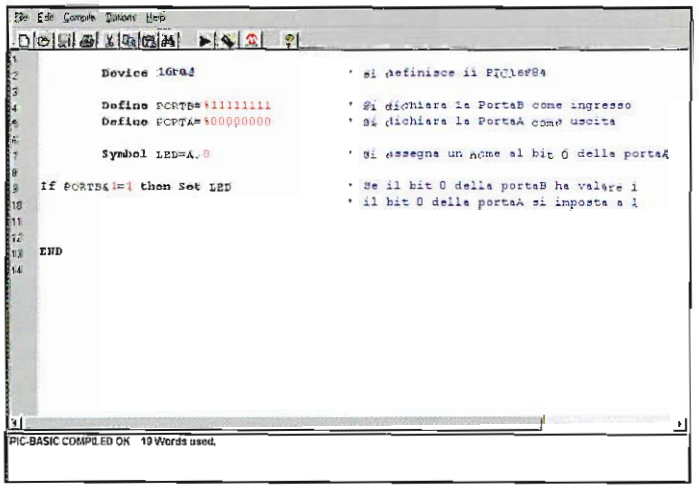
Verifica del valore di un bit

Immaginiamo che sia necessario verificare il valore di un bit di una porta, per cambiare il valore di altri bit in funzione del primo. La prima cosa che ci può servire è utilizzare la struttura IF...THEN per realizzare la verifica, sia nominando direttamente il bit, sia utilizzando il comando "SYMBOL" per dare un nome simbolico a questo bit. Allo stesso modo, per assegnare il valore a un secondo bit in funzione del valore del primo, lo potremo fare direttamente sul bit oppure sul nome che gli avremo assegnato.

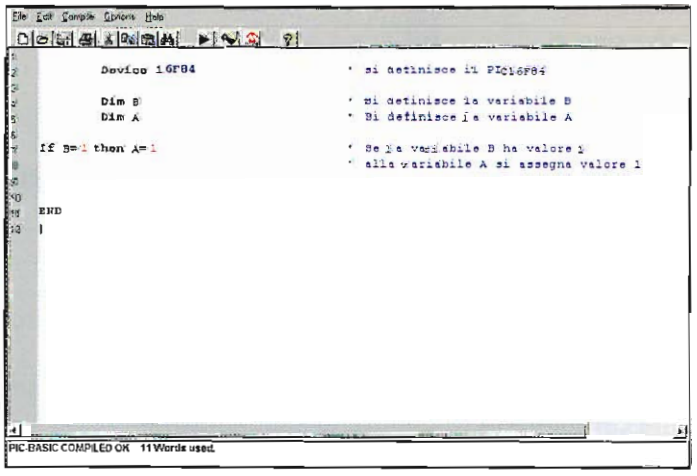
Nel primo esempio, si è



L'idea più intuitiva per il lavoro con i bit non è corretta.



Bisogna utilizzare istruzioni specifiche per il lavoro con i bit.



Il lavoro con i registri ammette queste espressioni.

cercato di fare proprio questo. Provate anche voi e guardate che cosa succede: compilando il programma appaiono diversi errori.

L'accesso a un bit non si può realizzare in questo modo, né tanto meno si può fare così l'assegnazione di un valore a un bit. Lavorando con i dati delle porte di ingresso/uscita abbiamo visto che dopo aver raccolto il valore completo della porta, è possibile utilizzare una maschera per "far sparire" i bit che non ci interessano. Questa procedura è utile sia per conservare un solo bit, che per diversi bit. Anche in questo caso assegneremo dei valori zero e uno a questi bit individualmente, direttamente oppure mediante il comando "SYMBOL", per dare un nome simbolico a ogni bit. Cambiando queste due idee nella struttura IF...THEN il compilatore non presenta alcuna obiezione.

Resta da dire però che la prima struttura, nel caso del lavoro con i registri, è corretta.

Attesa di un valore

Continuiamo a lavorare con i singoli bit. Pensate ora a un sistema che non inizi a funzionare sino a che non si prema un pulsante. Potremmo utilizzare ciò che abbiamo visto sinora per fare entrare il programma in un ciclo sino a che non si compia la condizione, così come si può vedere nella figura. Tenete conto che nell'esempio è stato considerato che il pulsante sia collegato per livello alto. Nel caso in cui sia collegato per livello basso bisogna cambiare la condizione per cercare il valore contrario; quindi a seconda se colleghiamo il LED sul livello alto o basso, mandando un "1" con l'istruzione "set" lo accenderemo oppure lo spegneremo.

Esiste anche un altro modo per realizzare questa operazione, cioè mediante l'istruzione "button". Quando trova un'istruzione di questo tipo il programma si ferma, e attende che il pin specificato cambi

```
Device 16F84                    ' si definisce il PIC16F84

Define PORTB=%11111111          ' Si dichiara la PortaB come ingresso
Define PORTA=%00000000          ' Si dichiara la PortaA come uscita

Symbol LED=A.0                  ' Si assegna un nome al bit 0 della portaA

Loop:   If PORTB&1=0 THEN goto Loop   ' Sino a che il bit 0 della PortaB ha
                                      ' valore 0 si rimane nel ciclo

        set LED                 ' Quando passa a valore 1 si continua
                                ' il programma e il LED si imposta a 1

END
```

PIC-BASIC COMPILED OK 20 Words used.

Il programma rimane in un ciclo sino a che non si preme il pulsante.

```
Device 16F84                    ' si definisce il PIC16F84

Define PORTB=%11111111          ' Si dichiara la PortaB come ingresso
Define PORTA=%00000000          ' Si dichiara la PortaA come uscita

Symbol INT=B.0                  ' Si assegna un nome al bit 0 della portaB
Symbol LED=A.0                  ' Si assegna un nome al bit 0 della portaA

Button INT                      ' Il programma non continua sino a che
                                ' INT non cambia di valore

Set LED                         ' Il bit 0 della PortaA si imposta a 1

END
```

PIC-BASIC COMPILED OK 16 Words used.

Il programma si ferma sino a che non è premuto un pulsante.

```
Device 16F84                    ' si definisce il PIC16F84

Define PORTB=%11111111          ' Si dichiara la PortaB come ingresso
Define PORTA=%00000000          ' Si dichiara la PortaA come uscita

Symbol INT=B.0                  ' Si assegna un nome al bit 0 della portaB
Symbol LED=A.0                  ' Si assegna un nome al bit 0 della portaA

Button INT
Set LED                         ' Il bit 0 della portaA si imposta a 1

Button B.1
Set A.1                         ' Il bit 1 della portaA si imposta a 1

Button B.3
Clear A.4                       ' Il bit 4 della portaA si imposta a 0

END
```

PIC-BASIC COMPILED OK 26 Words used.

Verifica di diversi interruttori con l'istruzione "button".