

## Strutture di controllo

Tutti i linguaggi di programmazione hanno bisogno di strutture di controllo per permettere che il flusso del programma non sia sequenziale, ma che prenda differenti percorsi a seconda dei registri delle variabili di ingresso, ecc. Tuttavia non tutti i linguaggi hanno queste strutture dichiarate in forma esplicita; questo è appunto il caso dell'assembler dei PIC, in cui le strutture di controllo si formano a partire da istruzioni "normali".

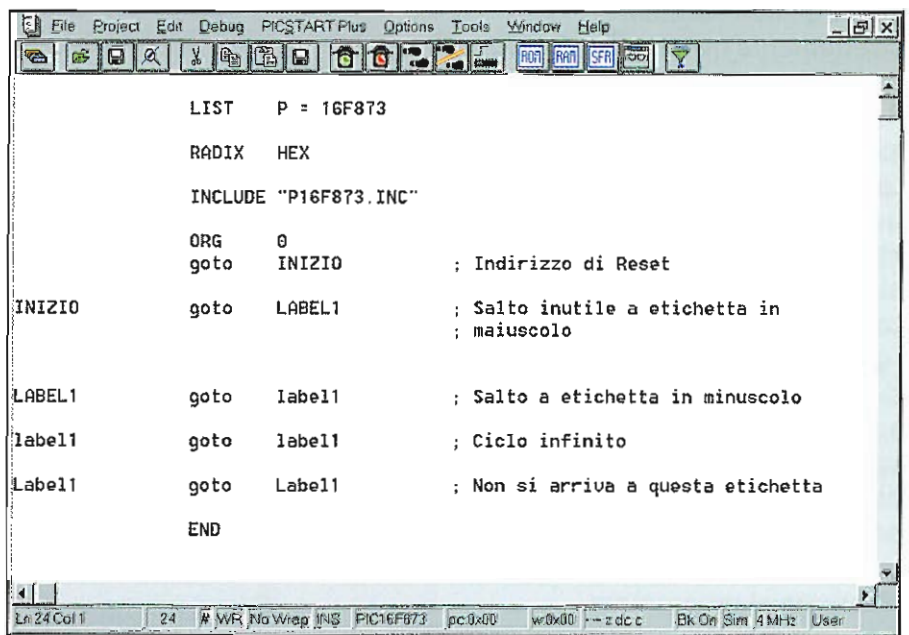
Questo ha i suoi vantaggi e i suoi inconvenienti, in quanto semplifica il numero di istruzioni da conoscere e utilizzare, però bisogna aver chiaro il concetto della struttura di controllo che si vuole ottenere.

### Salto incondizionato

Per rompere la sequenza di un programma la prima istruzione che esiste nell'assembler è la "goto". Questa istruzione devia il flusso delle istruzioni verso l'indirizzo puntato dal nome dell'etichetta, che è l'unico parametro dell'istruzione. Viene detto "incondizionato" perché l'istruzione funziona da sola e non dipende da nessuna condizione per realizzare il salto. Salta sempre. Insieme ad altre istruzioni, però, serve per realizzare strutture di controllo più complesse; inoltre, nell'assembler dei PIC non si

MNEMONICO	PARAMETRI	SIGNIFICATO
goto	etichetta	Salta all'indirizzo puntato da "etichetta"

Operazioni di salto incondizionale.



```
LIST P = 16F873
RADIX HEX
INCLUDE "P16F873.INC"
ORG 0
goto INIZIO ; Indirizzo di Reset
INIZIO goto LABEL1 ; Salto inutile a etichetta in
; maiuscolo
LABEL1 goto label1 ; Salto a etichetta in minuscolo
label1 goto label1 ; Ciclo infinito
Label1 goto Label1 ; Non si arriva a questa etichetta
END
```

Bisogna fare attenzione a utilizzare l'istruzione "goto".

potrebbero realizzare strutture di controllo senza la collaborazione di questa istruzione. Nell'assembler non è possibile sconsigliarne l'utilizzo, tuttavia non è strano che lo si faccia in altri linguaggi di programmazione, dove esistono già altre strutture di controllo più complesse. Bisogna quindi fare molta attenzione al momento dell'utilizzo.

Nell'esempio possiamo

osservare che sono state definite diverse etichette praticamente uguali dove cambiano solo le lettere maiuscole o minuscole all'interno del nome.

L'assembler dei PIC si può configurare perché sia "sensibile" a questi cambi o perché li consideri uguali. Possiamo verificare come risponde il programma cercando di compilarlo. Se si produce un errore nella compilazione,

significa che sono state prese tutte le definizioni di etichetta come uguali e dato che non è possibile definire un'etichetta più di una volta, questo è considerato un errore. Anche se non si producono errori questo, il programma non è corretto semanticamente, in altre parole "non ha senso".

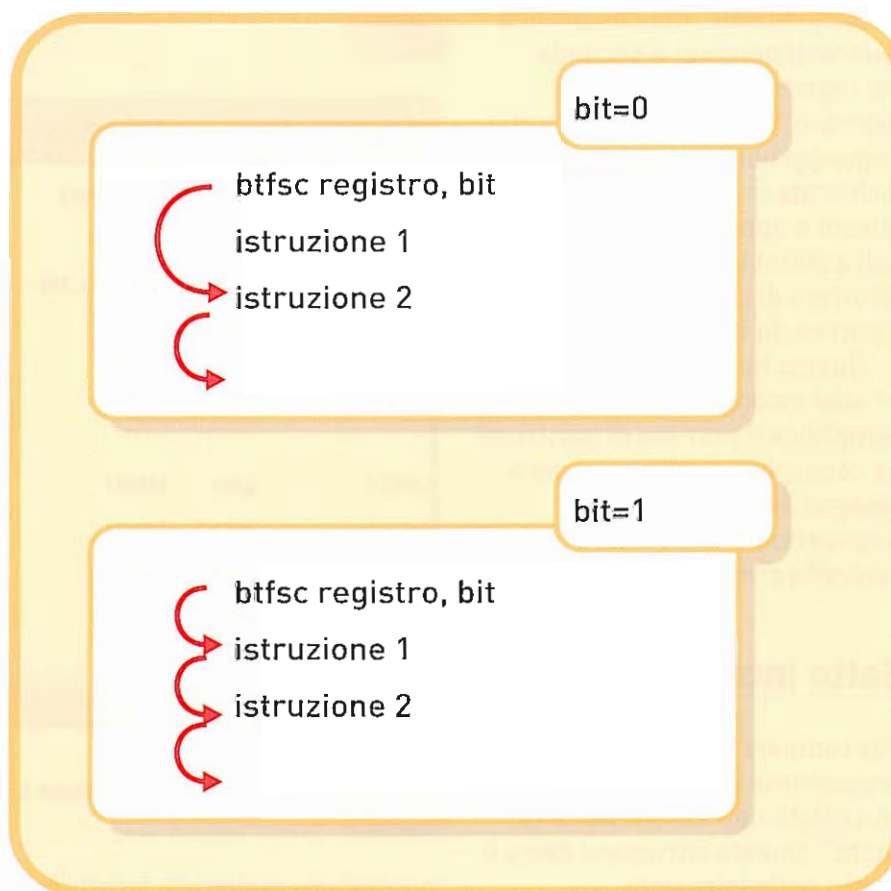
Normalmente dopo la direttiva "ORG 0" troviamo un'istruzione "goto INIZIO", anche se non sarebbe strettamente necessaria. Questo perché è fondamentale che il punto di inizio del programma sia chiaro, in quanto teoricamente prima e dopo queste istruzioni potrebbero essere inserite ancora altre routines. Tuttavia in altri casi l'inserimento di istruzioni "goto", quando l'etichetta di salto indica la casella successiva, non serve altro che a sprecare spazio nella memoria delle istruzioni, già di per sé scarsa. Bisogna dire però che questa ridondanza aiuta a dare chiarezza al programma. È importante fare attenzione ai punti morti come quelli della figura precedente, da cui non è più possibile fare nulla. Se arrivare a determinate etichette non dipende da nessuna condizione e si sa già a priori che non saranno utilizzate, non ha senso indicarle. Dato che oltre a occupare spazio extra, disorientano anche chi, cerca di capire il programma.

## Salti dipendenti da un bit

Nell'assembler del PIC, grazie all'utilizzo delle due istruzioni

MNEMONICO	PARAMETRI	SIGNIFICATO
btfsc	F, B	Salta un'istruzione se il bit B del registro F ha valore zero.
btfss	F, B	Salta un'istruzione se il bit B del registro F ha valore uno.

Operazione di salto condizionale.



Schema di funzionamento delle operazioni di salto condizionale.

presentate nella tabella in alto, combinate con la "goto" è possibile realizzare tutti i tipi di strutture di controllo. In realtà sarebbe già sufficiente una delle due, dato che entrambe verificano delle condizioni complementari, però così risulta molto più comodo. Anche

se i nomi di queste istruzioni sembrano dare poche informazioni su ciò che fanno, analizzando lettera per lettera, vediamo che questo non è vero. La "b" fa riferimento a un bit, la "t" significa "test", cioè verifica di un valore; "f" fa riferimento ai nomi dei registri in questo