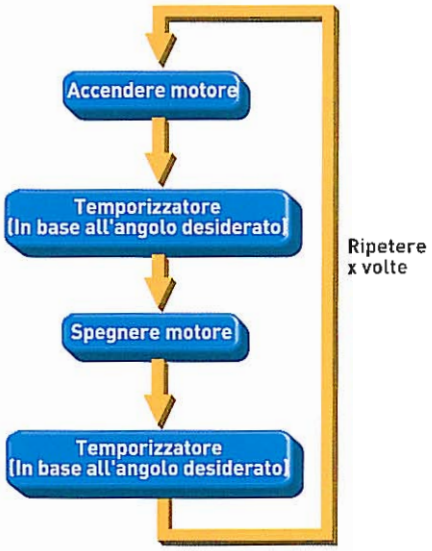
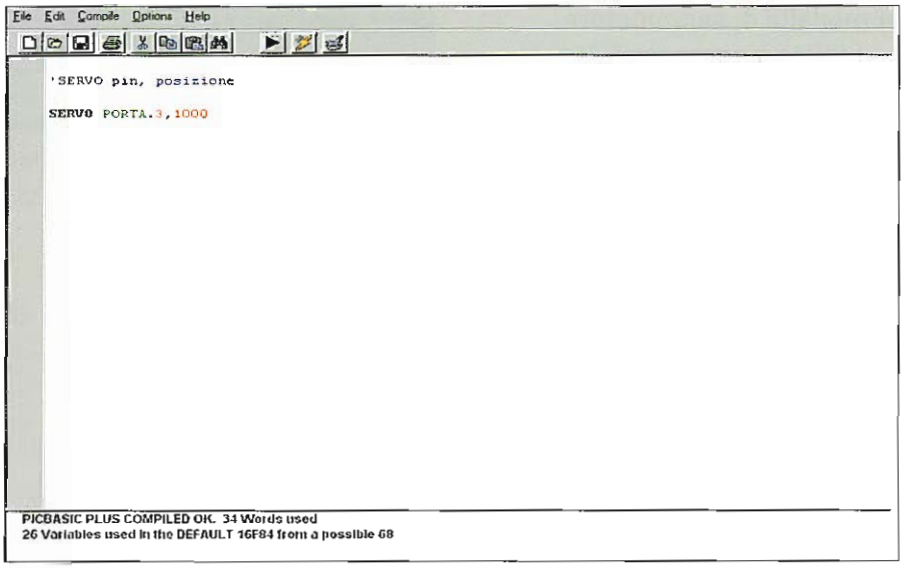


Organigramma per la gestione di un servomotore con istruzioni "generali".



Esempio di utilizzo dell'istruzione SERVO.

sull'angolo corretto, il motore si ferma, fino a quando verrà applicato un altro treno di impulsi di differente ampiezza.

Gestione di un servomotore in modo "manuale"

Ci sono linguaggi di programmazione che non sono preparati per il controllo diretto di un servomotore, dato che non hanno istruzioni specifiche a questo scopo. Questo, ad esempio, è anche il caso degli assembler dei PIC, in cui per gestire un servomotore bisognerà seguire un organigramma come quello riportato nella figura, simulando i treni di impulsi con accensioni e spegnimenti del motore con una temporizzazione variabile, in base all'angolo desiderato.

Nel migliore dei casi alcuni microcontroller, come quelli della famiglia 16F87x, dispongono di un dispositivo interno (CCP) per realizzare la modulazione di ampiezza degli impulsi (PWM), il che suppone di conoscere come lavora questo modulo, come si configura, quali sono i suoi registri, ecc. per cercare di ottenere l'obiettivo con l'utilizzo di istruzioni "generali".

Istruzioni di gestione di un servomotore

In linguaggi come il LetPicBasicPlus, invece, la gestione dei servomotori è molto semplice, dato che esiste una istruzione specifica per tale uso. L'istruzione SERVO genera un impulso della durata pari ai microsecondi specificati nel secondo parametro, sul piedino specificato come primo parametro, così come mostra l'esempio della figura.

La risoluzione dell'istruzione è sempre di 1 microsecondo, indipendentemente dal quarzo utilizzato nel microcontroller. Anche se il valore impostato può variare fra 0 e 65.535, normalmente varia tra 500 e 2.500, valori che di solito corrispondono agli angoli estremi.

Per generare un treno d'impulsi e fare in modo che l'asse del motore si muova alla posizione richiesta, è necessario ripetere questa istruzione un determinato numero di volte, avendo cura di lasciare fra un impulso e l'altro il tempo necessario per ottenere la frequenza di 50 impulsi al secondo. Nell'esempio è stato inserito un Delay di 20 millisecondi, però questo valore può variare in base

al modello di servomotore.

Inoltre è necessario tenere presente che l'istruzione DELAYMS è influenzata dal tipo di quarzo utilizzato.

Nel programma della figura l'asse del motore inizierà a muoversi dalla posizione centrale.

Programma esempio di gestione di un servomotore

Come con qualsiasi attuatore è possibile realizzare un'applicazione in cui il valore di questo dipenda da un dispositivo di ingresso. Immaginate, ad esempio, di volere orientare un'antenna montata su un servomotore secondo i valori di due pulsanti. Si partirà dal centro.

L'utente manterrà uno dei due pulsanti premuto per portare l'antenna a un estremo, o attiverà l'altro per portarla all'estremo opposto. Se non si preme alcun pulsante il servomotore rimarrà nell'ultima posizione.

È necessario aggiungere la sezione di inizializzazione del programma.

Una variante su tale programma potrebbe consistere nell'inserire tramite degli interruttori un valore per muovere l'asse del motore a un determinato angolo, senza la necessità di premere continuamente il pulsante.

Anche in questo caso il servomotore rimarrà in questa posizione sino a nuovo ordine.

File Edit Compile Options Help

```
' Si ripete continuamente il treno di impulsi. Arrivato alla posizione
' indicata, il servomotore non si muoverà più

loop: SERVO PORTA.3,1500
      DELAYMS 20
      GOTO loop
```

PICBASIC PLUS COMPILED OK. 55 Words used
26 Variables used in the DEFAULT 16F84 from a possible 68

Generazione di un treno di impulsi, utilizzando l'istruzione SERVO.

File Edit Compile Options Help

```
DIM pos AS BYTE

'Si parte da una posizione centrale
pos=1500
loop:
     ' Se si preme il pulsante del piedino 0 si incrementa la posizione
     IF PORTB.0=1 THEN IF pos<2000 THEN pos=pos+1

     ' Se si preme il pulsante del piedino 1 si decrementa la posizione
     IF PORTB.1=1 THEN IF pos<1000 THEN pos=pos-1

     ' Si applica al servomotore il valore della posizione
     SERVO PORTB.2,pos
     DELAYMS 20

     GOTO loop
```

PICBASIC PLUS COMPILED OK. 64 Words used
27 Variables used in the DEFAULT 16F84 from a possible 68

Esempio di programma di utilizzo di un servomotore.