

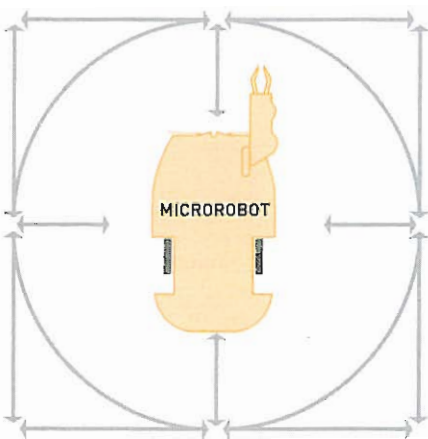
# Il microrobot ballerino

L'esecuzione di un determinato compito composto da diversi passi, implica la preventiva acquisizione di questi passi e la conoscenza dell'ordine specifico che segue. Questo serve sia per il montaggio di un'auto, che per lo sviluppo di una coreografia di ballo.

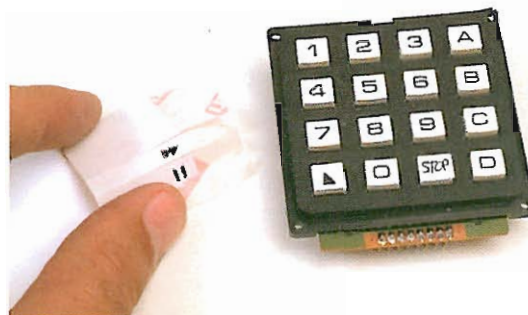
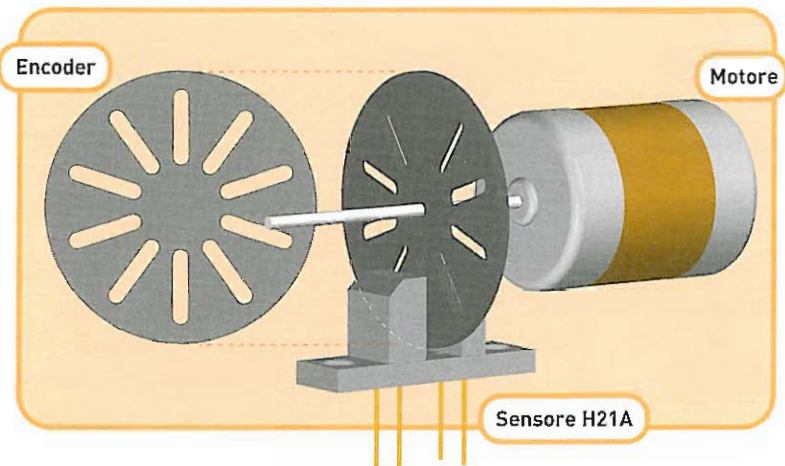
In entrambi i casi è necessario possedere delle conoscenze in memoria, e avere la possibilità di utilizzarle in sequenze differenti, in funzione dell'ambiente.

## Sensori

I passi che potrà eseguire il microrobot saranno scritti nella memoria, per fare questo non è necessaria la memoria EEPROM ma sarà sufficiente iniziarli a ogni inizio di programma come etichette o che siano definiti come macro o subroutine. Abbiamo bisogno di un modo per indicare la sequenza che desideriamo in ogni



Possibile schema di movimenti con due unici motori.



Sensori utilizzati per il microrobot ballerino.

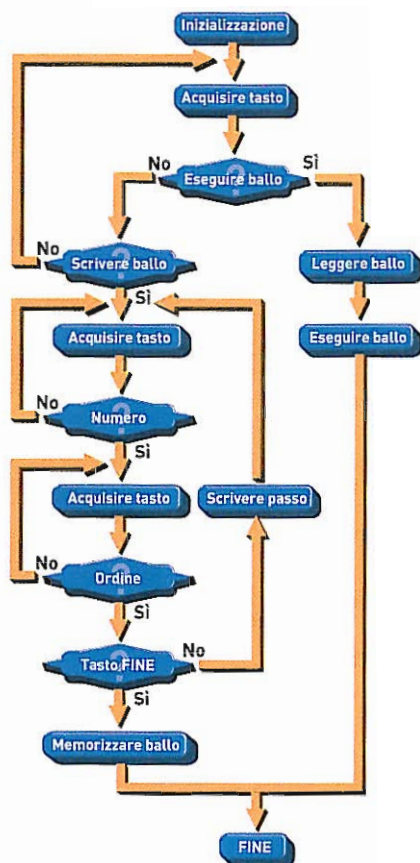
momento, ovvero il ballo in sé. Per questo ci servono due tipi di dati, numeri e ordini, per poter eseguire coreografie del tipo "due passi avanti, due passi indietro, rotazione di 90° ecc.". Dato che questi balli si possono cambiare, si può memorizzare la loro sequenza nella EEPROM, indicando ogni volta se si vuole eseguire un ballo già memorizzato o memorizzarne uno nuovo. Una buona scelta è utilizzare una tastiera esadecimale, i dieci numeri (da 0 a 9) possono servire per indicare il numero di passi, le lettere per l'azione scelta. Lo 0 si può prendere come numero 10 o come un tempo in cui non fare nulla. Nel caso in cui si

desiderassero più di 10 passi si potrebbe attendere l'attivazione consecutiva di due numeri.

La stessa concatenazione si può realizzare fra le lettere per ottenere più azioni. Sui tasti sarà possibile porre degli adesivi che aiutino a ricordare le funzioni. Dato che i movimenti si misurano in passi, sarà necessario anche un sensore con la funzione "encoder" per calcolare il numero di rotazioni della ruota.

## Motori

Come minimo saranno necessari i motori di movimento, anche se



ricevere è se deve eseguire un ballo già memorizzato o attendere la ricezione di un ballo nuovo. La tastiera è gestita da una routine che realizza continuamente una scansione, sino a quando non viene premuto un pulsante, a quel punto fornisce come risposta un codice diverso per ognuno dei pulsanti, identificandoli. Nel caso i tasti non siano né quelli di esecuzione né quelli di scrittura, il programma si fermerà in un ciclo. È anche possibile impostare per default che se il pulsante premuto non è di scrittura, lo si consideri di esecuzione, oppure viceversa.

Per scrivere un nuovo ballo si acquisiranno ad uno ad uno i pulsanti, considerando che il primo è il numero di volte da eseguire il passo, e il secondo è il passo in se stesso. Abbiamo già commentato che potrebbero essere coppie di pulsanti invece di uno solo, sia per la prima che per la seconda funzione. Un'altra opzione è che tutti i tasti servono per entrambe le cose a seconda se vengono attivati per primi o per secondi. Bisogna riservare anche un pulsante per segnalare la fine dei passi da scrivere. Ogni passo si scriverà sulla memoria RAM, solo alla fine il ballo completo verrà scritto sulla EEPROM. Allo stesso modo quando si vuole eseguire il ballo la cosa

migliore è leggere la EEPROM per spostare tutte le informazioni relative ai passi sulla RAM e da qui eseguirli completamente. Le ragioni per scrivere prima sulla RAM e solo alla fine sulla EEPROM, e poi viceversa sono due: da un lato abbiamo già visto che il lavoro con la EEPROM non solo potrebbe creare dei problemi, ma ha bisogno di più tempo, in questo modo invece non si rallenta l'acquisizione dei dati né lo sviluppo di un ballo una volta iniziato. Però ancora più importante di questo, è il modo con cui si può scrivere l'informazione sulla EEPROM. Il vero problema nel nostro caso è la sua capacità: se scriviamo i dati tali e quali come ci vengono forniti dalle routine della tastiera, potremo solo memorizzare 32 coppie N°- ordine, dato che ogni dato è identificato con un byte. Questo risulta insufficiente se vogliamo ottenere una forma di ballo un po' vistosa. Tuttavia è possibile duplicare lo spazio a disposizione in un modo semplice, per farlo è sufficiente "tradurre" queste informazioni in modo che ogni numero e ogni ordine occupino solo 4 bit. Al momento di eseguire il ballo è meglio tornare a dividere gli indirizzi, in modo che numeri e ordini occupino byte separati, perché questo renderà più facile il loro trattamento.

**Organigramma di scrittura o esecuzione di un ballo da parte del microrobot.**

tutti sappiamo che non è la stessa cosa ballare un waltzer in cui praticamente intervengono solamente i piedi rispetto a un ballo sivigliano in cui il movimento delle braccia costituisce già di per sé un passo. Tutto dipenderà, quindi, dal realismo che si vorrà dare, tuttavia non è necessario avere molti motori per poter fare tanti passi differenti; inoltre con un solo ordine sarà possibile dar luogo a diversi passi concatenati.

## Processo da seguire

Il microrobot comunicherà con l'esterno tramite la tastiera, e il primo dato che si attende di

TASTO	CODICE ROUTINE	CODICE 4 BIT
0	1011 1110	0000
1	0111 0111	0001
2	1011 0111	0010
3	1101 0111	0011
4	0111 1011	0100
5	1011 1011	0101
6	1101 1011	0110
7	0111 1101	0111
8	1011 1101	1000
9	1101 1101	1001
A	0111 1110	1010
B	1101 1110	1011
C	1110 1110	1100
D	1110 1101	1101
E	1110 1011	1110
F	1110 0111	1111

**Traduzione di quanto acquisito tramite la tastiera con codici da 4 bit.**