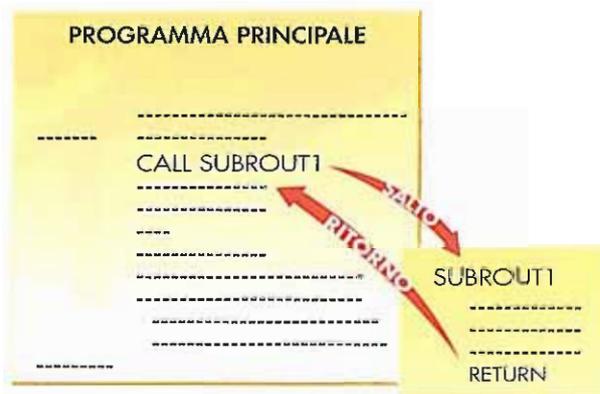


Funzionamento degli interrupt

Come indica il suo nome, un interrupt è uno stop momentaneo del flusso di controllo del programma principale motivato da una causa speciale. Quando si produce qualche causa speciale, se il processore accetta gli interrupt, salva il valore del PC sullo stack e carica il PC con il Vector di Interrupt che corrisponde all'indirizzo 0004 Hex della memoria delle istruzioni.

In questo indirizzo troviamo la prima istruzione di una routine dedicata all'interrupt e alla causa che lo ha provocato. Quando termina l'esecuzione della routine dedicata all'interrupt, si ritorna al programma principale dal punto dove lo si è abbandonato, recuperando dallo stack il valore per il PC.

Un interrupt è una rottura "asincrona" del flusso di



In funzione di quale antenna del microrobot rileverà l'ostacolo si realizzerà la routine di interrupt corrispondente.

programma, perché non si conosce il momento in cui avviene. Le cause che la provocano sono esterne al programma. Questa caratteristica distingue gli interrupt dalle "chiamate a subroutine", che sono originate dall'esecuzione di un'istruzione CALL all'interno del programma.

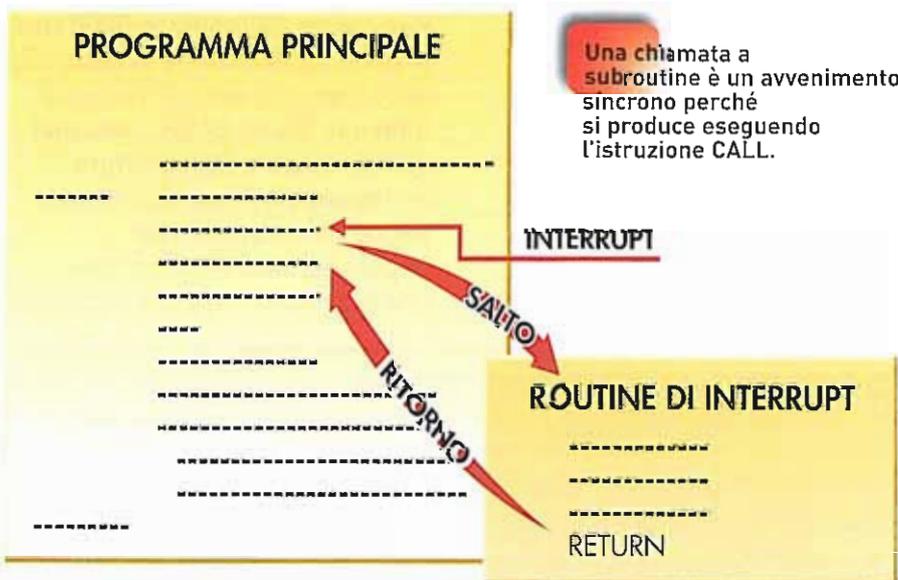
Anche in questo caso si scrive

il valore del PC sullo stack e si salta a realizzare una routine per ritornare al programma principale quando quest'ultima è terminata, però è un avvenimento "sincrono" perché si conosce il momento in cui si produce.

Applicazione degli interrupt

Una possibile causa di richiesta di interrupt è l'applicazione di un livello, o di un fronte, specifico su uno dei piedini del microcontroller. Mediante gli interrupt si risponde in modo immediato alle esigenze del mondo esterno. Nel microrobot a sei zampe della figura della pagina successiva ci sono due antenne che rilevano gli ostacoli sui lati della traiettoria.

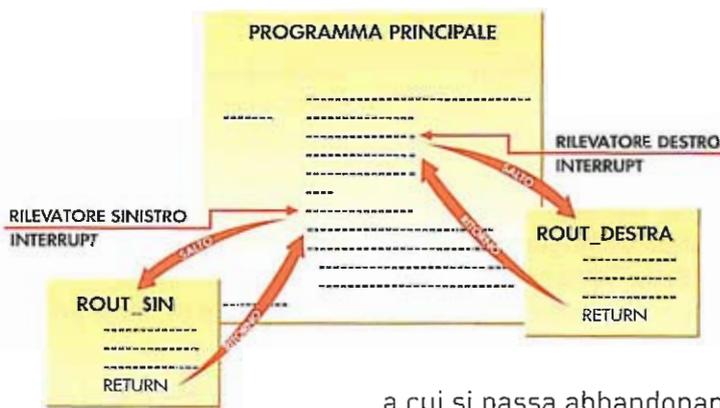
Quando a un certo punto della traiettoria l'antenna sinistra rileva ed entra in contatto con un ostacolo,



Una chiamata a subroutine è un avvenimento sincro perché si produce eseguendo l'istruzione CALL.



Le due antenne del microrobot rilevano gli ostacoli ai lati della traiettoria e generano un segnale a cui viene data immediata risposta producendo un'azione atta a schivare l'ostacolo.



Quando si accetta un interrupt, il programma principale si ferma fino al completamento della routine dedicata all'interrupt.

a cui si passa abbandonando il programma principale è RUT_DESTRA, come possiamo vedere nella figura.

Il Vector di Interrupt unico

Esistono numerose cause che possono provocare un interrupt,

però nei PIC si utilizza sempre lo stesso Vector di Interrupt, indipendentemente però dalla causa che lo ha generato.

Esso corrisponde all'indirizzo 0004 Hex della memoria. Per questa ragione il primo lavoro della routine di interrupt, che inizia sempre all'indirizzo 0004 Hex è verificare quale è stata la causa che lo ha provocato per poi saltare alla subroutine adeguata, ovviamente dovremo avere a disposizione una subroutine per ogni possibile causa. Un altro problema che è legato al Vector di Interrupt è rappresentato dal Vector di Reset che corrisponde all'indirizzo 0000 Hex, cioè occupa un indirizzo posto quattro posizioni prima.

Visto che il programma principale deve iniziare all'indirizzo 0000 Hex, arrivati all'indirizzo 0004 Hex bisogna saltarlo altrimenti finirebbe nella routine di interrupt. Normalmente nel Vector di Interrupt si colloca un'istruzione di salto (GOTO), a un indirizzo dove inizia lo spazio libero dedicato alla routine di interrupt. Per contro all'indirizzo 0000 Hex si pone un'istruzione di salto alla 0005 Hex, che evita il Vector di Interrupt. È una prassi comune, quindi, iniziare i programmi in linguaggio Assembler con le seguenti istruzioni che implementano le operazioni che vi abbiamo appena descritto.

ORG 0		; È un comando di Assembler che mette il PC = 0
GOTO	INIZIO	; Istruzione di salto all'etichetta INIZIO
ORG 5		; Comando che imposta il PC = 5
INIZIO	CLRWDT	; Istruzione che occupa la posizione 5 della memoria