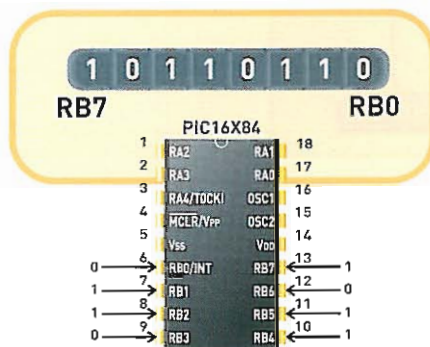


Programma che compara i valori di ingresso

Nel corso dell'opera abbiamo già visto come il robot abbia bisogno di sensori in grado di fornirgli le informazioni necessarie a prendere le decisioni. Ogni singola informazione, pur essendo importante, può non essere sufficiente per prendere una decisione, a volte può essere necessario comparare diversi valori d'ingresso.

Presentazione del problema

Il problema si presenta così ampio da dover essere delimitato. Ci sono tre casi principali, secondo se stiamo lavorando con interrupt, valori digitali, o valori analogici. Nel caso degli interrupt i sensori di ingresso sono digitali, e generano un interrupt quando si produce un fronte, o un cambio di livello sull'ingresso, secondo il tipo di interrupt. Nei PIC tutti i tipi di interrupt hanno la stessa priorità, quindi la decisione verrebbe determinata dall'interrupt che si produce per primo fra tutti quelli abilitati, o fra il valore in ingresso



Una combinazione di dati digitali crea la funzione di un valore analogico.

da un pin normale e uno che può produrre interrupt, dato che quest'ultimo ha la priorità sul precedente. Se lavoriamo senza interrupt abbiamo la possibilità di avere ingressi digitali o analogici.

Un ingresso digitale può avere solo due valori, 0 o 1, per cui la comparazione si baserebbe unicamente su quale dei diversi sensori si è attivato, e nel caso ve ne fosse più di uno dovremmo assegnare delle priorità per decidere chi "comanda". Tuttavia la combinazione di diversi ingressi digitali può simulare il funzionamento di un valore analogico, se si comparano diversi dati relativi allo stesso

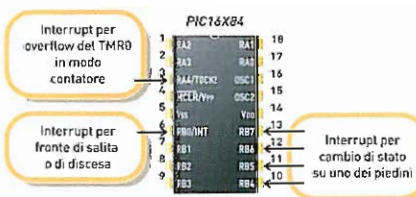
tempo. Per ultimo possiamo lavorare con diversi sensori analogici, e mediante un convertitore esterno, o il convertitore analogico-digitale che hanno alcuni PIC, trasformando il loro dato in digitale per poterlo elaborare normalmente.

Idea dell'algoritmo

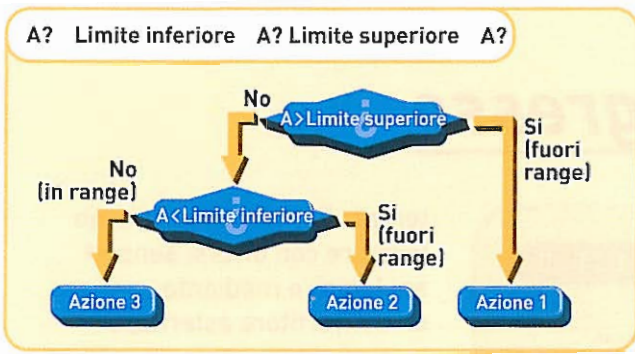
Dimentichiamo per un momento il caso degli interrupt, e concentriamoci sui valori analogici e sui valori digitali composti, che alla fine si riducono alla stessa cosa. La comparazione può essere fatta per uguaglianza, per maggioranza o per minoranza. In tutti i casi l'organigramma è lo stesso.

La differenza la ritroveremo al momento di decodificare l'organigramma in uno o nell'altro linguaggio di programmazione. Nessun metodo è più complesso di un altro, l'utilizzo dipenderà dai motivi per cui viene scelto. Un caso molto tipico è stabilire se un determinato valore si trova all'interno di un range, in questo caso entrano in gioco tre valori: i

Piedini del PIC che possono ricevere interrupt.



Organigramma di comparazione fra variabili.



Determinazione della pertinenza ad un range.

(8 bit meno significativi), e l'algoritmo risulterà ancora più complesso, soprattutto se ciò che si sta facendo è determinare l'appartenenza a un range. In alcuni casi, il linguaggio utilizzato contribuisce a complicare le comparazioni anche se i dati hanno la dimensione di soli 8 bit.

In un linguaggio con strutture di controllo come il BASIC è sufficiente utilizzare una struttura

limiti inferiore e superiore e la variabile da comparare.

Nella figura è riportato un possibile organigramma.

Sensori e attuatori

L'algoritmo presentato è valido sia per valori digitali composti che per quelli analogici.

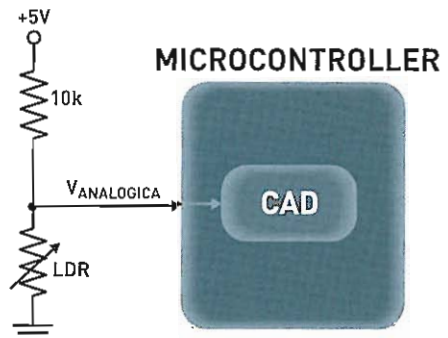
Nel caso dei valori digitali, se i numeri da comparare non arrivano a occupare un registro completo (8 bit) bisognerà impostare a zero i bit non utilizzati, e questi dovranno essere i bit più significativi del registro (quelli di sinistra), in caso contrario potremmo fare delle comparazioni errate.

I valori analogici, tuttavia, non arrivano su una sola linea, e sarà il convertitore (interno o esterno) che avrà il compito di "tradurli" in un valore di X bit, dove X può essere 8 (data la facilità di gestione dei dati di questa dimensione), oppure un valore maggiore; tanto maggiore sarà il numero dei bit, maggiore sarà la precisione del valore. Per quanto riguarda gli attuatori, tutto dipenderà da ciò che si vuole fare dopo aver risolto la comparazione, poiché non fanno parte dell'algoritmo.

Restrizioni

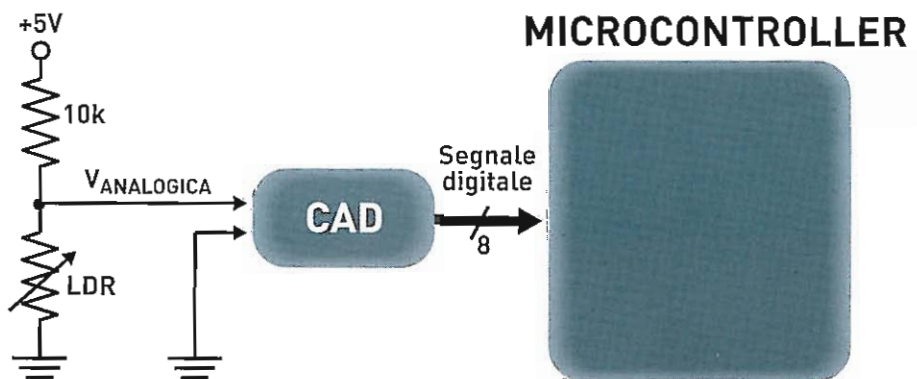
L'idea dell'algoritmo è facile da capire e non presenta ambiguità, per cui non si può parlare di restrizioni propriamente dette. Tuttavia la traduzione dell'organigramma a programma può essere più o meno problematica secondo gli elementi di cui si dispone, e del linguaggio di programmazione.

Ad esempio se il valore digitale diretto, o convertito da quello analogico, è minore della dimensione del registro, bisognerà "riempire" quest'ultimo sino ad arrivare a 8. Se è maggiore di 8 la comparazione si complica, dato che bisognerà tener conto sia della parte alta (più significativa) del dato sia della parte bassa



Sensore analogico collegato a un PIC con convertitore interno.

IF...THEN...ELSE, però nel linguaggio assembler dei PIC queste strutture non esistono, anzi in realtà non esistono proprio le strutture di controllo, quindi la traduzione non è così immediata. Tratteremo questo argomento più avanti, con le istruzioni assembler.



Sensore analogico collegato al PIC tramite un convertitore esterno.

