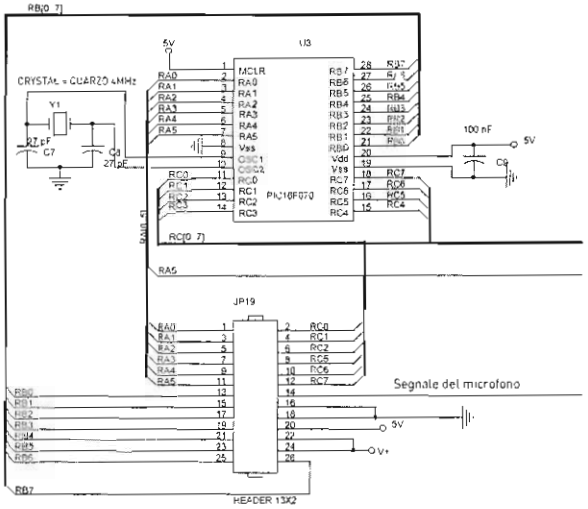
Strumenti da utilizzare

Abbiamo già parlato dei programmi software che utilizzeremo nel lavoro con i microrobot. La programmazione però non è tutto; se vogliamo che le applicazioni a cui stiamo pensando siano facili da realizzare e possano essere modificate anche dopo un certo tempo, è necessario essere molto sistematici, e seguire sempre le stesse procedure.



Gli schemi elettronici possono essere più o meno complessi a seconda dell'applicazione.

Definizione del problema

La prima cosa da sapere è cosa vogliamo fare, in seguito ci preoccuperemo di come farlo. Immaginate di dover spiegare la vostra applicazione a un'altra persona, per fare in modo che la capisca: in questo caso un ottimo metodo potrebbe essere elencare una serie di cose che sia capace di fare il microrobot.

VOGLIO UN MICROROBOT CHE:

√ Segua un percorso.

√ Se incontra un ostacolo si ferma e fa suonare un cicalino sino a che l'ostacolo non viene rimosso.

√ Arrivando ad un bivio sceglie un percorso oppure l'altro a seconda se la temperatura sia maggiore o minore di 22°C

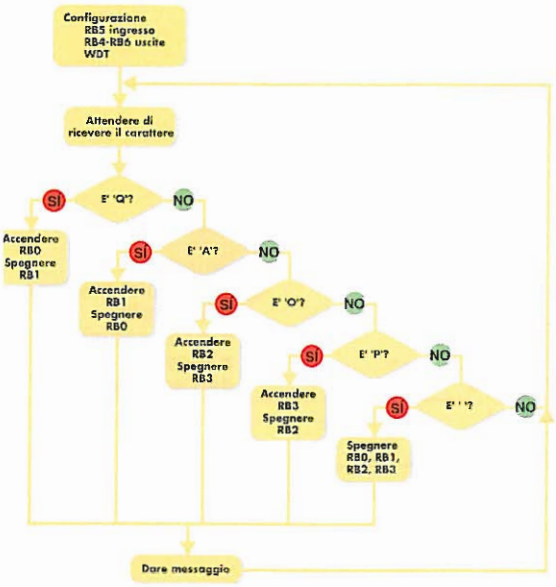
Dobbiamo aver chiaro ciò che vogliamo fare.

Schema elettronico

Abbiamo appena enunciato la nostra applicazione con frasi semplici, però che tipo di materiale sarà necessario per costruirla? Se si tratta di un microrobot inseguitore, avrà bisogno di due motori per muoversi, e di due sensori CNY70 per riconoscere il percorso. Per rilevare gli ostacoli dovrà avere almeno un sensore meccanico, come ad esempio un fine-corsa, e un cicalino che suoni ad ogni urto. Infine dovrà avere anche un sensore di temperatura per scegliere il percorso adeguato nel caso di un bivio.Tutte queste idee si rappresentano in quello che si chiama schema elettronico. Gli schemi elettronici possono essere più o meno complessi, secondo l'applicazione. Nel nostro caso avremo il PIC come parte centrale, e collegheremo ai suoi piedini gli elementi hardware necessari. Lo scopo dello schema elettronico è, nella fase di definizione del problema, assicurare che non esistano conflitti nell'utilizzo dei piedini, e al momento della programmazione quello di trovare a prima vista i piedini utilizzati.

Organigramma

Quando abbiamo parlato dei "concetti generali" abbiamo già spiegato in che cosa consistono gli organigrammi, e le regole generali a cui devono sottostare. Per verificare che l'organigramma corrisponda alla definizione del problema, chiediamo a qualcuno che ci spieghi che cosa capisce leggendo l'organigramma stesso. Si possono realizzare diversi organigrammi, uno generale e molto semplice, e altri per entrare nel dettaglio (alcuni dei rettangoli) dell'organigramma generale; si tratta di trasformare un'idea astratta in qualcosa di concreto.



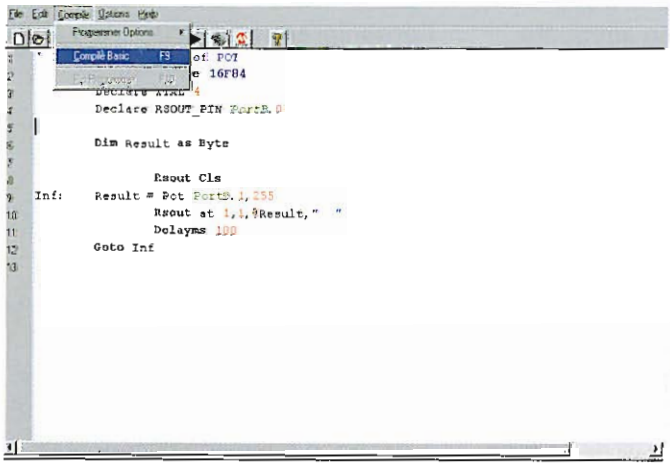
Un organigramma può risultare più o meno complesso.

Programma commentato

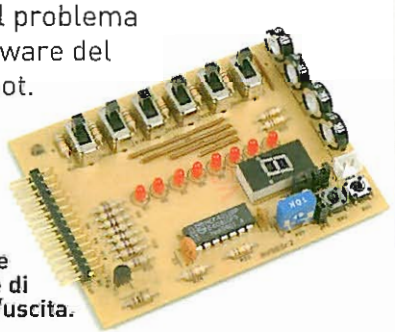
Avendo ora a disposizione lo schema elettronico e l'organigramma, si tratta di stendere il programma. Il linguaggio dipenderà dalla fase di apprendimento in cui ci troviamo; normalmente ogni linguaggio possiede il proprio editor. Bisogna fare la traduzione dell'organigramma pezzo per pezzo, cioè tradurre in istruzioni l'enunciato di un rettangolo, poi quello del successivo e così via, utilizzando le strutture di controllo a seconda delle alternative (rappresentate dai rettangoli) ecc. Se l'organigramma è stato costruito bene, non sarà molto difficile tradurlo in istruzioni. Lo schema elettronico ci aiuterà a fare in modo che ogni piedino sia collegato alla periferica a cui si fa riferimento nel programma.
Non dimenticate i commenti, vi aiuteranno a capire il programma anche dopo molto tempo.

Messa a punto

Se abbiamo già il programma è il momento di provarlo. La prova si sviluppa in diverse fasi, per prima cosa dobbiamo assicurarci di non aver commesso errori di sintassi, che sono quelli immediatamente individuati nel momento in cui si cerca di compilare il programma, e sono causati da parole scritte male, assenza di parametri ecc. Dopo aver corretto questo tipo di errori dobbiamo verificare gli errori semantici, che si producono quando il programma non fa quello che ci aspettiamo. Quando si lavora con software e hardware allo stesso tempo, questi problemi possono essere causati da una non adeguata programmazione o dal difetto di una qualche parte hardware, come il microcontroller, i sensori, ecc. In questi casi la cosa migliore è eseguire delle verifiche pezzo per pezzo.
L'idoneità del programma di solito si prova con l'ausilio di un simulatore, come MPLAB, che impareremo a utilizzare.
Dobbiamo ricordare, inoltre, che i simulatori non ricreano al cento per cento la stessa condizione ambientale in cui verrà fatto girare il programma, per cui quando il software verrà scritto sul microcontroller potrebbero emergere dei difetti, che nella simulazione non erano presenti. Si rende pertanto necessario l'impiego di alcuni strumenti di utilizzo generale, come ad esempio una scheda di ingressi/uscite con cui eseguire delle verifiche. Per utilizzare questi strumenti bisogna verificare che gli elementi di ingresso/uscita siano collegati come previsto dallo schema elettronico. Dopo aver verificato il software, non ci rimane che scriverlo sul microcontroller finale e provare il microrobot. Se a questo punto emergeranno ancora dei difetti, sapremo di dover cercare il problema nell'hardware del microrobot.



Dopo aver scritto il programma bisognerà compilarlo.



È utile impiegare le schede di ingresso/uscita.