

BARCOLLO... MA NON CROLLO!



dell'ing. MIRCO SEGATELLO



Da fermo o in movimento sta in equilibrio sulle sue due ruote. È il nostro nuovo robot, un piccolo Segway con cui sperimentare le problematiche hardware e software che coinvolgono la realizzazione di un mezzo con un solo asse...

Non è certo la prima volta che parliamo di robot, però un automa come quello descritto in queste pagine siamo certi che non l'abbiate mai visto; ma non aspettatevi un androide in kit di montaggio o chissà cosa: non vedrete schede elettroniche strapiene di componenti, né strutture meccaniche particolarmente elaborate, ma solo due ruote, un piccolo circuito, una batteria e nulla di più. E allora, direte voi, dove sta la particolarità di questo robot? Per spiegarlo è necessario fare un passo indietro, a quando i robot erano "scatole ambulanti" e compivano i loro primi movimenti: la principale difficoltà era farli muovere e restare stabili, in piedi; in un primo tempo si cercò la soluzione più stabile, che consisteva nel farli muovere su tre o più ruote. Poi si passò ai primi umanoidi, che si spostavano su gambe; il "robot bipede" che abbiamo presentato nel fascicolo numero 133 è di questa categoria e può contare su due piedi di generose dimensioni che gli permettono di conservare l'equilibrio sia in movimento

che da fermo. Proprio con questi robot ci si pose il problema di farli rimanere in equilibrio, problema che sta tutto nel mantenerne il baricentro entro la base d'appoggio, cosa che può essere fatta controllando opportunamente i movimenti, ovvero evitando quelli che porterebbero il baricentro fuori dalla base d'appoggio.

Ebbene, il robot qui descritto è particolare perché funziona al contrario: se restasse fermo cadrebbe di sicuro. Eppure sta su due sole ruote, ondeggia, si sposta qua e là, ma non cade mai! Diversamente dal bipede, questo resta in equilibrio compiendo continui movimenti, che gli servono a mantenere il baricentro entro la base d'appoggio, la quale si identifica con la linea che unisce le due superfici di contatto con il suolo.

Potete capire la logica del problema guardando la Fig. 1, che raffigura un umanoide e due robot su ruote: tutti hanno il centro di gravità all'interno dei punti di appoggio. Anche senza conoscere tutte le leggi della Fisica, possiamo ipotizzare

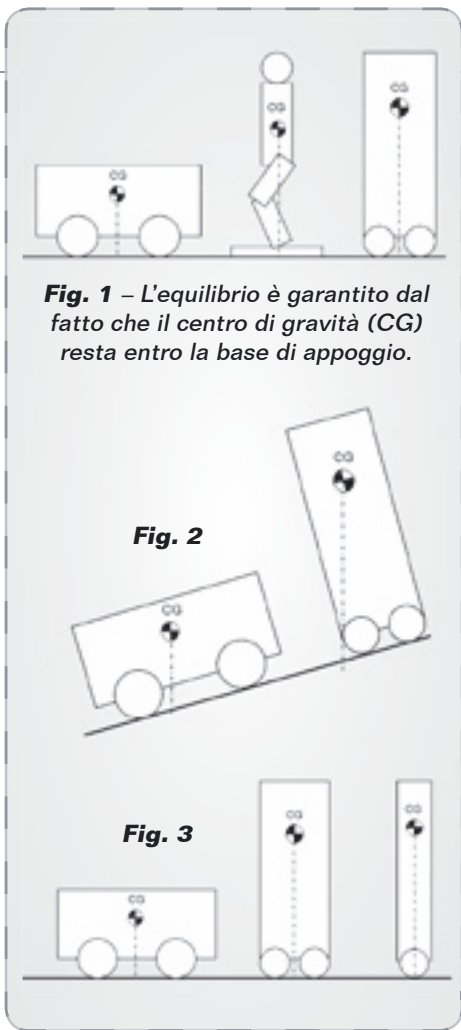


Fig. 1 – L'equilibrio è garantito dal fatto che il centro di gravità (CG) resta entro la base di appoggio.

Fig. 2

Fig. 3

che più il baricentro è alto, più la base di appoggio è stretta e meno stabile risulterà la struttura. Basterebbe chiedersi quale struttura è più facile da rovesciare e subito verrebbe in mente quella più a destra della figura. Immaginiamo adesso cosa accadrebbe se il piano su cui poggiano i robot fosse inclinato. Nella struttura a destra il baricentro fuoriesce dalla base di appoggio causando, inevitabilmente, la caduta (Fig. 2).

Tutto questo ci fa capire che strutture alte e strette per loro natura mantengono difficilmente una condizione di stabilità. Tuttavia se ragioniamo in ambito robotico possiamo dedurre che strutture alte possono meglio interagire con l'uomo: un robot che vi porge una bibita a 20 cm da terra lo trovereste comodo? Inoltre una base di appoggio ristretta permette di muoversi agevol-

mente in ambienti domestici. Ecco perché ultimamente si sta studiando la stabilità dei robot e come essa possa essere gestita. Noi che non siamo da meno ci siamo voluti cimentare anche in questo campo, rivolgendo la nostra attenzione verso un caso particolare, estremo, che prevede una base di appoggio infinitesima e configura quindi la situazione di assoluta instabilità. Partendo dal sostegno a ruote a due assi, immaginate di avvicinare sempre più le ruote di un lato fino a farle coincidere: arriverete ad avere un unico punto di appoggio (Fig. 3).

Essendo la struttura tridimensionale, in realtà i punti di appoggio saranno due (ruota destra e ruota sinistra) ma il concetto rimane invariato. Una struttura siffatta è ovviamente instabile per sua natura e tenderà a cadere. Il nostro intento è rendere la struttura stabile ed evitare che cada anche se dovessimo far inclinare la base su cui si appoggia. Ecco svelata la particolarità di questo progetto, il quale ricalca quel mezzo di trasporto elettrico americano noto come *Segway* (www.segway.com) che consiste in una sorta di carrello a un solo asse con manubrio e pedana sulla quale montare, capace di restare sempre in equilibrio e tenere dritto il passeggero sia fermo che in movimento.

APPROCCIO AL PROBLEMA

Per comprendere quanto sia difficile ottenere l'equilibrio, fate un piccolo esperimento: prendete un ombrello o un qualsivoglia oggetto lungo e stretto sufficientemente alto, ponete un'estremità su di un dito e provate a farlo rimanere in verticale senza farlo cadere. Dovreste riuscirvi senza grosse difficoltà, magari muovendo un po' il dito.

Ma che cosa succede in pratica? Ebbene, i vostri occhi osservano l'ombrello e se questo si muove avvisano il cervello, il quale invia alle mani i comandi per evitare che esso cada.

Provate adesso a fare la stessa cosa con una penna: non ci riuscite! Perché? Semplice, la penna cade più rapidamente dell'ombrello ed il vostro cervello non è abbastanza veloce da compensare la caduta. Da questo semplice esperimento abbiamo già ricavato notevoli informazioni sull'aspetto fisico del problema ed abbiamo già sotto mano, anzi in punta di dito, la soluzione al problema. L'argomento infatti può essere letteralmente ribaltato osservando che un ombrello posto in punta di dito altro non è se non un pendolo inverso e non a caso alcuni progetti del genere prendono il nome di "Inverted Pendulum". Da questa considerazione e dalle formule che legano il moto di un pendolo, possiamo stimare rapidamente la velocità del nostro sistema e quindi dedurre quanto dovrà essere veloce il controllo della stabilità che dovremmo adottare. L'equazione del moto di un pendolo permette di determinare il

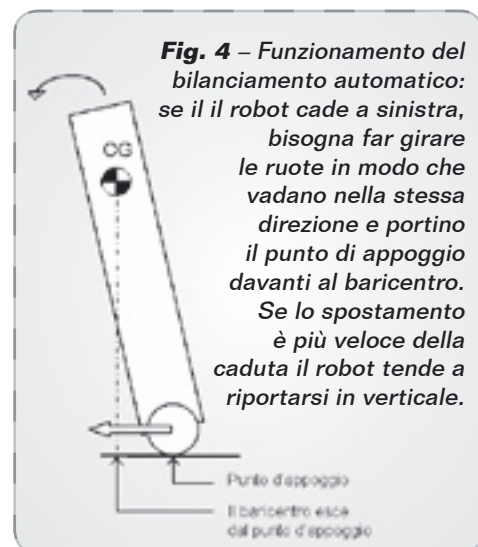


Fig. 4 – Funzionamento del bilanciamento automatico: se il robot cade a sinistra, bisogna far girare le ruote in modo che vadano nella stessa direzione e portino il punto di appoggio davanti al baricentro. Se lo spostamento è più veloce della caduta il robot tende a riportarsi in verticale.

periodo di oscillazione T in funzione della lunghezza l (distanza tra il fulcro e l'estremità inferiore dell'oggetto):

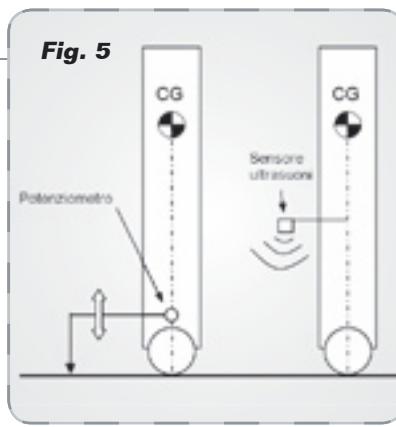
$$T = 2 \cdot \pi \cdot \sqrt{\frac{l}{g}}$$

dove g è l'accelerazione di gravità e vale tipicamente $9,81 \text{ m/s}^2$. Questo significa che l'oscillazione del pendolo dipende solo dalla sua lunghezza ma non dalle masse in gioco.

Per una struttura alta circa 40 cm come è il nostro robot, il tempo di oscillazione è calcolato in circa 1,3 sec; per intervenire e correggere in tempo la tendenza a cadere, dovremmo quindi essere almeno dieci volte più veloci. Sì, ma a fare che cosa? Questo è presto detto ricordando l'esperimento dell'ombrello, per non far cadere il quale bisogna spostare il dito che lo sostiene.

Per non far sdraiare il robot autobilanciante, dobbiamo mantenere il centro di gravità del robot esattamente in verticale sul punto di appoggio a terra e per farlo bisogna fare in modo che ogniqualvolta il corpo si inclina, la base si sposti da una parte o dall'altra fino a tornare sotto alla proiezione verticale del baricentro. Dunque, per far rimanere in piedi il nostro piccolo robot dobbiamo rilevare l'inclinazione mediante un apposito sensore, quindi comandare un motore elettrico che agisca sulle ruote per spostare in qua e in là la base. Il tutto, in un tempo dieci volte inferiore dei predetti 1,3 secondi. Ora va detto che il robot può rimanere bilanciato in verticale in diverse posizioni ma non necessariamente nella posizione iniziale, ma noi vorremmo anche che rimanesse dritto.

E' chiedere troppo?



IL SENSORE DI INCLINAZIONE

Come accennato, abbiamo bisogno di un sensore che ci permetta di determinare l'inclinazione del robot con notevole precisione e rapidità; tra le varie soluzioni ci sono l'utilizzo di un potenziometro solidale con la struttura che tramite una leva a contatto della superficie di appoggio varia la sua resistenza in funzione dell'inclinazione, oppure l'uso di un sensore ad ultrasuoni che misuri la distanza dell'apice del robot dalla superficie di appoggio (Fig. 4). Questi sistemi, peraltro validi, consentono di mantenere il robot perpendicolare al piano di appoggio e quindi sono assolutamente inadatti a funzionare su piani inclinati. La nostra soluzione, più raffinata, utilizza un accelerometro per rilevare l'inclinazione del robot rispetto alla posizione verticale.

Il componente utilizzato è l'MMA7260 della Freescale

Semiconductor prodotto che viene fornito premontato su una piccola basetta per consentirne l'utilizzo con normali connettori SIL a passo 2,54 mm; è composto da tre accelerometri basati sulla tecnologia a condensatore denominati *G-Cell*, che riescono a rilevare l'accelerazione alla quale esso è sottoposto. In condizioni statiche (cioè fermo) il sensore misura l'accelerazione di gravità nei tre assi fornendo un'indicazione sul suo orientamento nello spazio. Questo sistema viene usato anche negli smartphone di ultima generazione per modificare l'orientamento del contenuto del display a seconda di come si tiene in mano l'apparecchio. Quando il sensore si muove, le sue uscite forniscono una tensione proporzionale all'accelerazione cui è sottoposto in aggiunta all'accelerazione di gravità. La tensione a riposo (senza alcuna accelerazione) sulle tre uscite vale la metà di quella di alimentazione, cioè di 1,65 volt. Questa tensione sale se l'accelerazione avviene nel verso concorde alla direzione stampigliata sul sensore e diminuisce se è contraria. Scegliendo la maggior sensibilità ($\pm 1,5G$) la tensione in uscita avrà un range di $\pm 800mV$

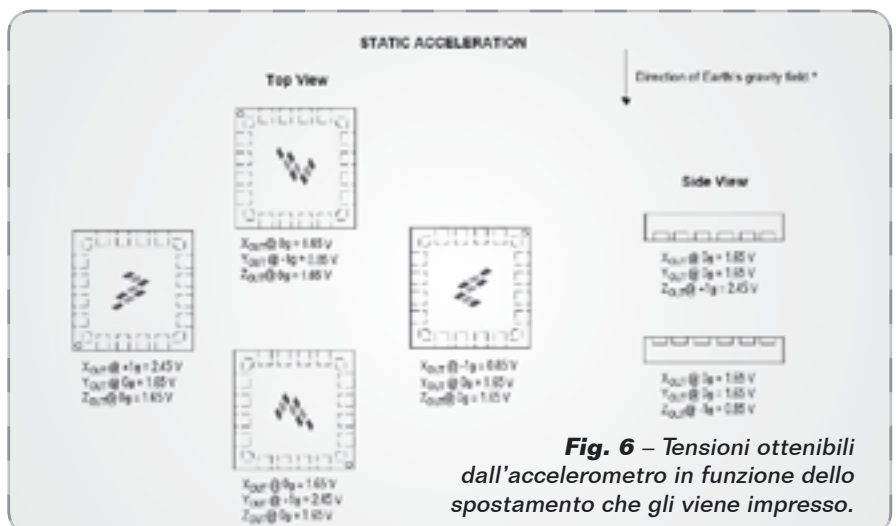


Fig. 6 – Tensioni ottenibili dall'accelerometro in funzione dello spostamento che gli viene impresso.



Fig. 7

quando il sensore sarà sottoposto ad un'accelerazione pari a $\pm 1G$. Ricordiamo che $1G$ rappresenta l'accelerazione di gravità. In condizioni statiche, ruotando il sensore

si ottengono le misure riportate nella Fig. 6.

Installando il modulo accelerometro sul nostro robot come mostrato nella Fig. 7, e utilizzando le due uscite relative all'asse X e Y possiamo risalire all'effettiva inclinazione della struttura rispetto alla direzione verticale. Per una misura assoluta sarebbe necessario ricorrere alla trigonometria, scrivendo la seguente equazione:

$$\alpha = \arctg\left(\frac{acc_Y}{acc_X}\right)$$

dove acc_X e acc_Y sono rispettivamente le due misure sui due assi depurate della tensione di offset (che in questo caso vale 1,65 V). Purtroppo implementare simili calcoli in un microcontrollore va fatto solamente se assolutamente necessario; nel nostro caso, non essendo necessario determinare in modo assoluto l'angolo e considerando che l'inclinazione massima del robot sarà solo di pochissimi gradi, optiamo per la misura nella sola direzione dell'asse X, per la quale vale l'equazione:

$$\alpha = Acc_X \cdot K$$

dove K è un'opportuna costante. Così facendo, ammettendo un'inclinazione entro i 10° rispetto alla verticale, l'errore massimo

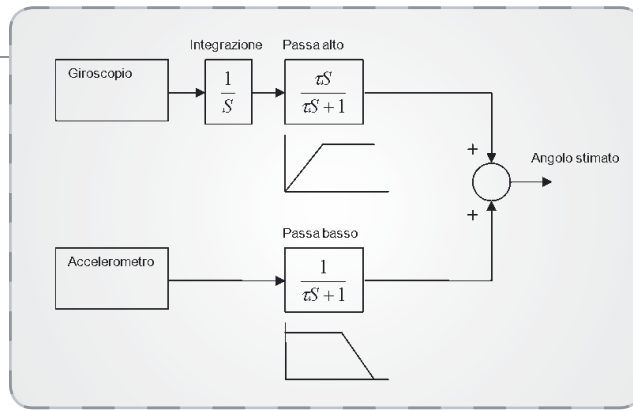


Fig. 8
Schema a blocchi e funzionale del filtro complementare.

che si commette è pari al 5%, valore più che accettabile per la nostra applicazione.

A questo punto, però, ci troviamo con due inconvenienti non da poco: il primo riguarda il fatto che la tensione fornita dall'accelerometro è afflitta da parecchio rumore, che dopo svariate misure eseguite sul componente in prova possiamo valutare in 30 mV in valore efficace e 120 mV come valore picco-picco. Un valore così elevato non assicura una misura molto precisa, se non eseguendo un calcolo di media aritmetica finalizzato ad estrapolarne il valore medio. Un secondo problema si presenta nella misura dinamica dell'accelerazione, affetta da un ritardo intrinseco dovuto al tipo di costruzione del componente. In termini pratici, ciò comporta un ritardo nell'informazione per cui, se il robot comincia a sbilanciarsi, l'intervento di correzione risulterà tardivo.

Per risolvere queste problematiche dobbiamo ricorrere ad un sensore più veloce, in grado di fornire delle indicazioni istantaneamente in modo molto "pulito"; a tal fine abbiamo scelto un giroscopio prodotto dalla Melexis, siglato MLX90609 e reperibile presso la ditta Futura Elettronica (www.futurashop.it). Il componente è per SMD e, analogamente all'accelero-

metro, viene fornito già saldato su un'apposita basetta, così da renderne facile l'utilizzo.

L'indicazione che ci fornisce questo sensore è relativa, nel senso che ci permette di valutare di quanti gradi si è ruotato il robot ma senza farci sapere esattamente di quanti gradi si trova inclinato. È questa la ragione per cui non possiamo usarlo da solo ma le sue indicazioni devono essere integrate da quelle fornite dall'accelerometro; lo capite meglio considerando le caratteristiche di ciascuno dei sensori.

L'accelerometro misura la gravità e consente di ricavare facilmente l'angolo di inclinazione assoluto; è preciso nella misura a lungo termine ma il segnale che fornisce è affetto da rumore e lento nell'aggiornarsi. Il giroscopio misura la velocità di rotazione, integrando la quale si ottiene l'angolo di rotazione, che però è relativo all'ultima posizione assunta e non assoluto. La misura ottenuta diventa imprecisa a lungo termine, però si adegua rapidamente alle variazioni. Per riuscire ad ottenere la misura dell'inclinazione del robot rispetto alla verticale l'unica soluzione è fondere le misure dei due sensori, sfruttando le loro caratteri-

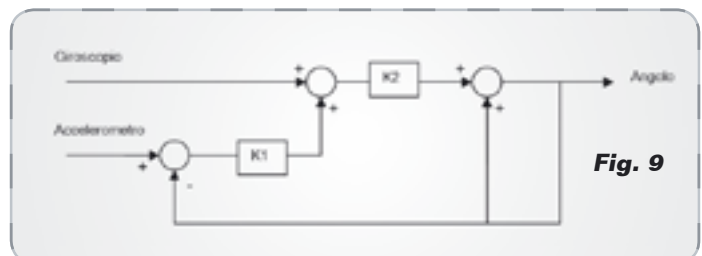
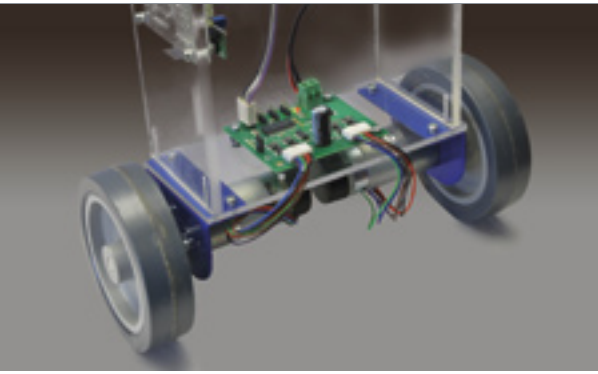
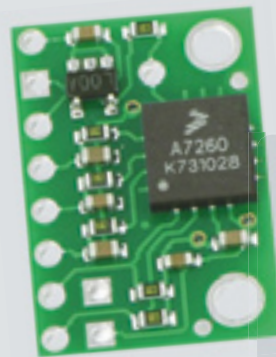


Fig. 9

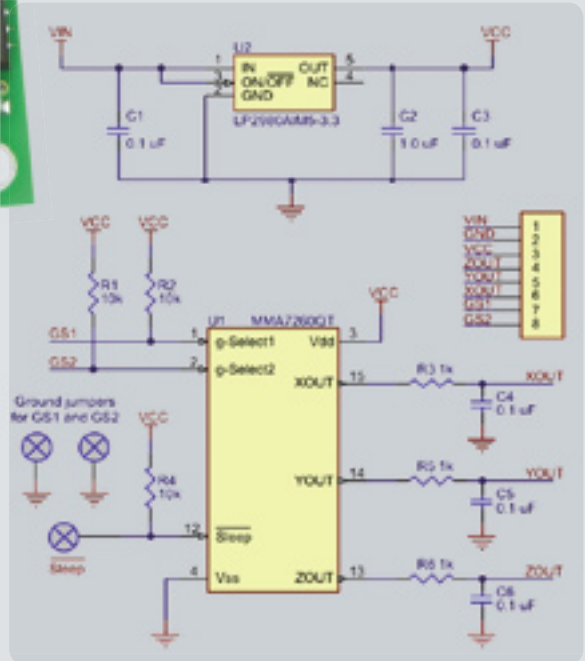


stiche migliori. Tra i vari sistemi a disposizione si trovano il *filtro di Kalman* e il *filtro complementare*. Il filtro di Kalman è sicuramente una valida soluzione, che però comporta una certa complessità matematica e un'implementazione non proprio facile, dovuta, questa ai molteplici coefficienti richiesti, non facilmente deducibili dalla caratteristiche reali dei componenti utilizzati. Il filtro complementare, almeno nella sua forma essenziale, è di facile implementazione e nello stesso tempo fornisce delle buone prestazioni; vediamo su cosa si basa. Si tratta, in sintesi (Fig. 8), di integrare il valore del giroscopio per ricavarne l'angolo, eliminando nel contempo la componente continua affetta dal *Drift* tramite un filtro passa-alto che lasci inalterate le caratteristiche dinamiche. A questa misura si somma il valore dell'accelerometro filtrato con un passa-basso che possa eliminare il rumore presente, lasciando inalterata la misura dell'angolo.

Esiste un'interessante implementazione del filtro complementare, che utilizza uno schema applicativo relativamente semplice, il cui schema a blocchi è rappresentato nella Fig. 9. Scegliendo le due costanti K1 e K2 tali che $K1 \cdot K2 = 1$, è possibile dimostrare che in condizioni statiche, quando il giroscopio fornisce valore nullo, il segnale di uscita è uguale a quello fornito dall'accelerometro. In condizioni dinamiche, con l'accelerometro



Il sensore di accelerazione è il chip l'MMA7260 Freescale che viene fornito già saldato su una piccola basetta ramata alla quale sono state aggiunte alcune resistenze e condensatori, oltre ad uno stabilizzatore di tensione. È sufficiente saldare lo strip a passo 2,54 mm



CARATTERISTICHE TECNICHE

- **Tensione di alimentazione:** 12 Vcc
- **Dimensioni:** 2 x 1.4 x 0,28 mm ;
- **Tensione di lavoro:** 3,3±16 V;
- **Corrente assorbita:** 1,35 mA;
- **Tensione d'uscita:** 0÷Vcc (0÷3,3 V per Vcc > 3,3 V);
- **Sensibilità:** ±1,5G/ ±2G/ ±4G o ±6G (selezionabile mediante i pin GS1 e GS2; ±6G è quella predefinita);
- **Peso:** 0,85 g.

e il piccolo modulo sensore è pronto per essere utilizzato. Per capire come funziona l'accelerometro osserviamo il suo schema elettrico, qui riportato, che comprende l'intero circuito contenuto nella basetta. Come potete vedere, pur essendo stato progettato per funzionare a 3,3 volt, l'accelerometro (grazie al regolatore integrato) può essere alimentato a tensioni anche maggiori, il che consente di impiegarlo in circuiti alimentati con i 5 V tipici dei dispositivi TTL. Le specifiche tecniche del modulo sono descritte a lato. I collegamenti esterni del modulo sono disposti su una fila di contatti a passo 2,54 e sono quelli elencati nell'apposita tabella.

PIN	NOME	FUNZIONE
1	Vin	Tensione di alimentazione (3,3 - 16V)
2	GND	Massa
3	Vcc	Tensione stabilizzata (3,3 volt)
4	Zout	Tensione di uscita riferita all'asse Z
5	Yout	Tensione di uscita riferita all'asse Y
6	Xout	Tensione di uscita riferita all'asse X
7	GS1	Selezione portata
8	GS2	Selezione portata

GS2	GS1	PORTATA
Aperto	Aperto	±1,5G
Aperto	Chiuso	±2G
Chiuso	Aperto	±4G
Chiuso	Chiuso	±6G

Qui sopra, impostazione della sensibilità dell'accelerometro mediante i piedini GS1 e GS2.

che fornisce valore nullo, l'uscita è l'integrale del valore fornito dal giroscopio. Questo sistema permette quindi di fondere i due sensori garantendo buone prestazioni sia in condizioni

statiche che dinamiche sfruttando le proprietà dei due sensori. A questo punto rimane il problema di determinare la posizione della base del robot rispetto alla linea di appoggio. Per fare ciò

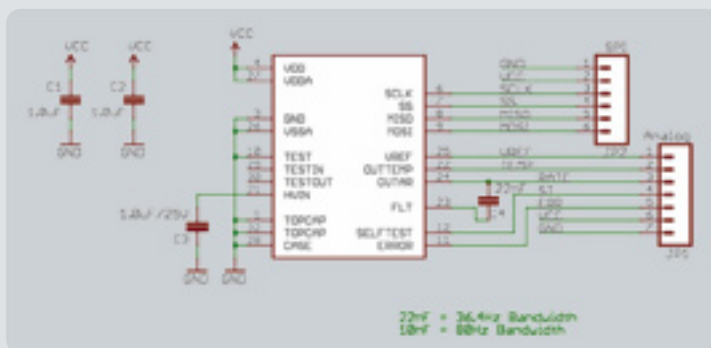
Il giroscopio

Dallo schema elettrico illustrato in questo riquadro possiamo vedere come nel modulo giroscopio siano stati usati pochissimi componenti oltre al chip (solo alcuni condensatori di disaccoppiamento). Il sensore si alimenta direttamente a 5 volt e l'uscita è analogica; non è richiesta alcuna impostazione hardware della sensibilità,

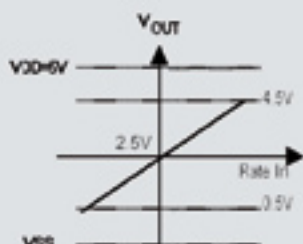
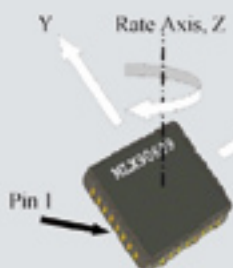
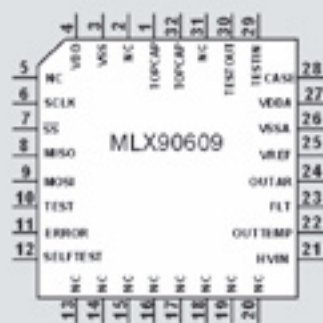
come nel caso dell'accelerometro. Il sensore dispone sia di un'uscita analogica che di un bus di comunicazione I²C adatto all'interfacciamento con i microcontrollori; nella nostra applicazione abbiamo usato l'uscita analogica. Su detta uscita, il sensore fornisce

un segnale analogico proporzionale alla velocità di rotazione rispetto all'asse Z, che è quello perpendicolare alle superfici più ampie del contenitore.

La tensione in uscita a sensore fermo è pari a metà di quella di alimentazione e cioè 2,5 V, ed aumenta o diminuisce a seconda della velocità di rotazione impressa al sensore (in senso orario o antiorario) sino



22nF = 36.4kHz Bandwidth
10kΩ = 88kHz Bandwidth



al limite della scala, che nel nostro caso è di ± 300 gradi/s (insomma, poco meno di un giro al secondo).

La principale applicazione del giroscopio non è quella di misurare la velocità di rotazione di alberi motori, bensì valutare la rotazione (la sterzata, ossia il cambio di direzione) dei veicoli: il suo impiego principale è quindi nella navigazione assistita da satellite.

Questo tipo di sensore fornisce un segnale "pulito" che segue molto rapidamente le variazioni della velocità di rotazione; presenta, però, alcuni limiti, primo fra tutti la deriva. Infatti il valore di offset a riposo di 2,5 volt risulta essere molto instabile ed influenzato sia da parametri ambientali (temperatura soprattutto) sia da una deriva nel tempo a causa della costruzione interna del compo-

nente. In inglese, per indicare questo fenomeno, viene usato il termine *Drift*. Per determinare l'angolo di rotazione è necessario eseguire l'integrale della velocità di rotazione secondo la seguente equazione:

$$\alpha = \int \text{gyro}(t) dt.$$

Si tratta di un'operazione che implica una serie di difficoltà nel momento in cui alla variabile gyro venga sovrapposto un segnale di offset non costante. In ogni caso, nella nostra applicazione l'angolo misurato non è quello di inclinazione del robot, in quanto la misura non è assoluta e può iniziare da una posizione qualsiasi; ricordiamo che se il robot è fermo il giroscopio fornisce un segnale nullo indipendentemente da quale sia la posizione in cui si trova.

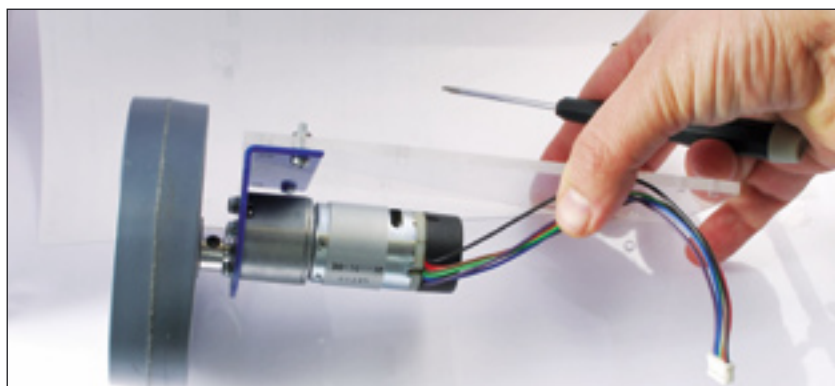
è indispensabile utilizzare un encoder posizionato sul motore adibito alla rotazione delle ruote in modo da misurare il numero di giri compiuto dalla ruota e di conseguenza lo spazio percorso. Contando il numero di impulsi dall'encoder si ottiene infatti una misura di distanza proporzionale al numero di giri compiuti dalla

ruota. Ovviamente l'encoder dovrà fornire non solo l'indicazione sullo stato della rotazione ma anche sul suo verso permettendo di stabilire anche la direzione di movimento ed il contatore sarà di tipo UP/DOWN.

IL GRUPPO DI PROPULSIONE

La trazione è affidata a due moto-

ri in corrente continua accessoriati di encoder. La scelta è caduta sul set RD01 della ditta Futura Elettronica, composto da due motoriduttori con encoder (EMG30), un driver di comando (MD23), due supporti (EMG30supp) e due ruote dal diametro di 100mm (WHEEL100). Il cuore del sistema di propulsione è il driver realizzato in tecnologia SMD che permette il controllo completo di due motori tramite comandi via bus I²C, i quali consentono la gestione della direzione e della velocità indipendente dai due motori, mentre in lettura è possibile leggere il valore dei contatori interni relativi agli encoder dei due motori. Poter variare indipendentemente la velocità dei motori consente di far compiere al robot anche spostamenti non



rettilinei; ciò serve, ad esempio, se volessimo farlo avanzare facendogli compiere delle curve. Nell'apposito riquadro riportiamo i dettagli del driver.

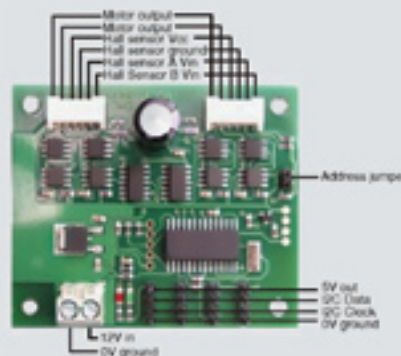
L'alimentazione prevista è di 12 V e sul bus di controllo possono essere prelevati 5 V stabilizzati (ricavati da un regolatore interno che può fornire 1 A di picco e 300mA continuamente) usati internamente dal driver e utili per alimentare un secondo circuito.

A parte i registri interni descritti nel box a lato, occupiamoci qui di una funzione molto importante: l'**Automatic Motor Timeout**, che, quando attiva, pone in "stop" entrambi i motori se non riceve sul bus alcun comando per più di due secondi.

Ciò serve ad evitare di veder correre inutilmente il robot se c'è un errore nella comunicazione con il circuito di controllo (il micro...) oppure, in applicazioni in cui si desidera radiocomandare il robot, quando quest'ultimo dovesse uscire dalla portata del comando a distanza.

SCHEMA ELETTRICO DEL CIRCUITO DI CONTROLLO

Veniamo ora alla descrizione dello schema elettrico della scheda di comando del nostro robot, la quale coordina l'attività dell'intero sistema dialogando con il controller MD23 e acquisendo le informazioni dell'accelerometro e del giroscopio. Come potete vedere, non vi è nulla di complesso, anzi la componentistica è molto scarna e tutta incentrata sul microcontrollore PIC18F2550. La scelta è caduta su questo chip non per l'interfaccia USB di cui dispone, bensì per la possibilità di lavorare con un clock di ben 48 MHz. Questa caratteristica è molto importante, visti i tempi stretti a disposizione e la mole di



Il driver è nato per funzionare tramite il bus standard I²C con indirizzo di default 0xB0, con la stessa modalità con la quale si comandano le memorie. L'indirizzo del driver può essere modificato a piacere, il che rende possibile la gestione di più dispositivi tutti connessi allo stesso bus. Internamente all'MD23 si trovano i registri elencati nella Tabella 1 che descriviamo dettagliatamente di seguito:

Speed1 Register

In modalità 0 o 1 questo registro imposta la velocità di rotazione del motore 1. In modalità 2 o 3 imposta la stessa velocità per entrambi i motori.

Speed2/Turn Register

In modalità 0 o 1 imposta la velocità del motore 2. In modalità 2 o 3 imposta la differenza di velocità tra il motore 1 ed il motore 2 (*Turn mode*) che permette la sterzata; sono possibili le seguenti combinazioni:

- motor speed1 = speed - turn;
- motor speed2 = speed + turn.

Oppure, compiendo la rotazione contraria:

- motor speed1 = speed + turn;
- motor speed2 = speed - turn.

Encoder registers

Ogni motore dispone di un encoder il cui conteggio avviene tramite un contatore interno a 32bit con segno. Il valore del contatore può essere ricavato leggendo separatamente i quattro byte da cui è composto.

Battery volts

Il valore della tensione di alimentazione è disponibile tramite questo registro. Il valore letto è decuplicato rispetto al valore reale. Ad esempio un valore di 121 corrispondono 12,1 V sulla batteria.

Motor 1 and 2 current

Il valore della corrente assorbita da ciascun motore è disponibile in questi registri. Il valore di corrente fornito è decuplicato rispetto al valore reale, ad esempio al valore 25 corrispondono 2,5 A sul motore.

Software Revision number

Il valore di questo registro corrisponde alla versione del firmware presente sul microcontrollore della scheda.

Acceleration Rate

Registro adibito al controllo dell'accelerazione dei motori. Il valore può essere impostato tra un minimo di 1 e un massimo di 10.

Mode Register

Questo registro imposta il modo di funzionamento:

0, Mode 1 (Default Setting) In questa modalità i valori impostati sui registri speed 1 e speed 2 controllano la velocità di rotazione in un range di: 0 (Full Reverse) 128 (Stop) 255 (Full Forward);

1, Mode 1 è simile al Mode 0, eccetto che nel range dei valori che valgono: -128 (Full Reverse) 0 (Stop) 127 (Full Forward);

2, Mode 2 entrambi i motori hanno la stessa velocità impostata con il registro speed1 con un range di: 0 (Full Reverse) 128 (Stop) 255 (Full Forward);

3, Mode 3 è simile al mode 2 eccetto nella scala dei valori che vale: -128 (Full Reverse) 0 (Stop) 127 (Full Forward).

Command register

Con questo registro possono essere impartiti i comandi esposti nella Tabella 2.

Changing the I²C Bus Address

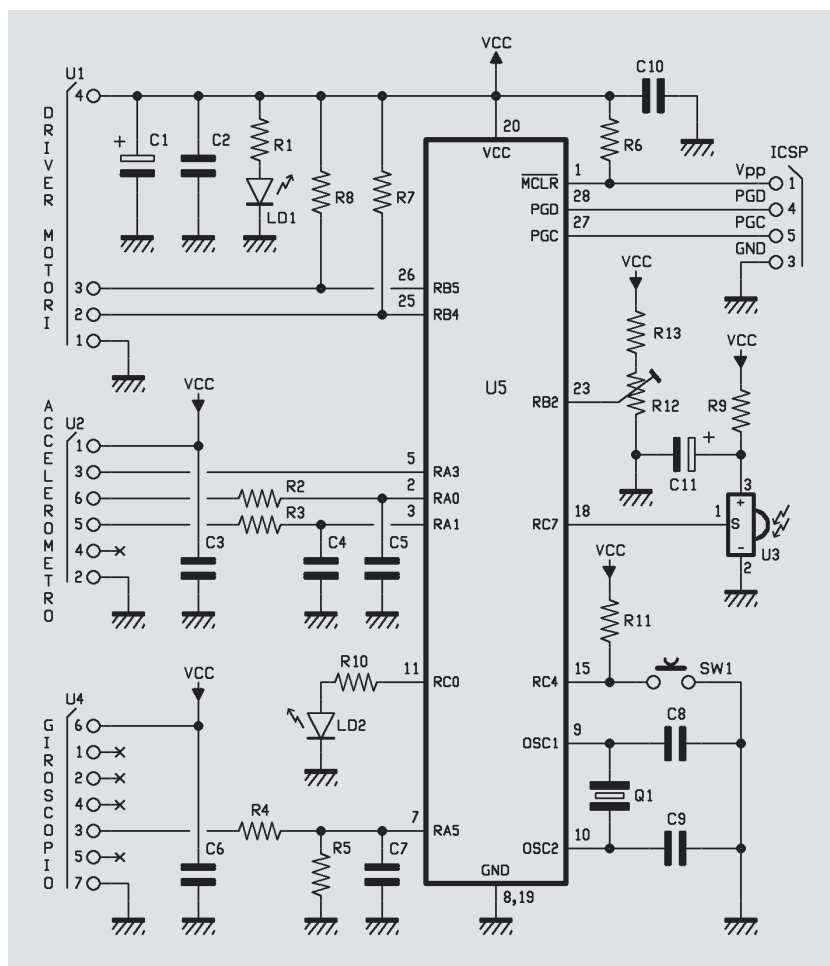
Riguarda l'impostazione dell'indirizzo che il modulo prenderà sul bus I²C; per la procedura dettagliata relativa al cambio di indirizzo si rinvia al manuale del driver.

Tabella 1 - I registri disponibili nel modulo MD23

Indirizzo	Nome	Funzione
0	Speed1	Velocità del motore1
1	Speed2/Turn	Velocità del motore2
2	Enc1a	Encoder1 1° byte
3	Enc1b	Encoder1 2° byte
4	Enc1c	Encoder1 3° byte
5	Enc1d	Encoder1 4° byte
6	Enc2a	Encoder2 1° byte
7	Enc2b	Encoder2 2° byte
8	Enc2c	Encoder2 3° byte
9	Enc2d	Encoder2 4° byte
10	Battery volts	Tensione sulla batteria
11	Motor1 current	Corrente sul motore 1
12	Motor2 current	Corrente sul motore 2
13	Software version	Versione del software
14	Accelerator rate	Accelerazione
15	Mode	Modo di funzionamento
16	Command	Comando

Tabella 2 - I comandi relativi al registro Command Register

VALORE (decimale)	COMANDO
32	Resetta i contatori degli encoder.
48	Disabilita la regolazione automatica della velocità
49	Abilita la regolazione automatica della velocità
50	Disabilita la funzione di TimeOut
51	Abilita la funzione di TimeOut
160	Prima sequenza per il cambio di indirizzo
170	Seconda sequenza per il cambio di indirizzo
165	Terza sequenza per il cambio di indirizzo



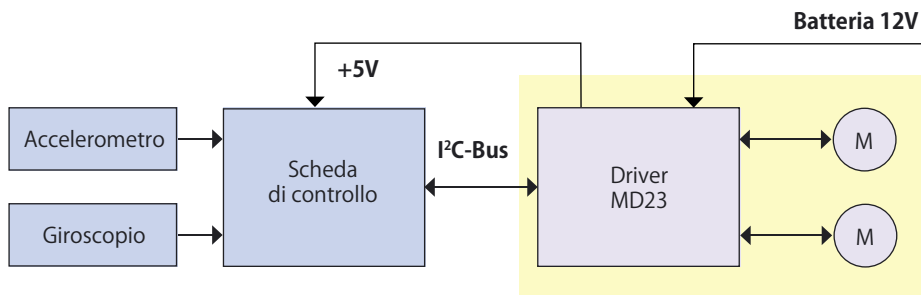
calcoli da sviluppare. Per il collegamento tra il driver MD23 e la basetta abbiamo previsto un connettore dal quale il circuito di controllo preleva la tensione di alimentazione stabilizzata a 5 volt; di questa connessione, le due linee SDA e SCL sono il bus I²C. ICSP è il connettore per la programmazione la cui pedinatura è compatibile con il programmatore della Microchip PICKIT2. Il LED D1 indica la corretta alimentazione. Altri due connettori vengono utilizzati per collegare l'accelerometro e il giroscopio: in entrambi i casi sfruttiamo la linea analogica. Per migliorare la lettura del segnale la linea Vcc relativa alla tensione stabilizzata di alimentazione interna all'accelerometro viene usata come riferimento di tensione per il convertitore A/D; le resistenze R4 e R5 formano un

partitore per adattare la tensione in uscita dal giroscopio (alimentato a 5V) ai livelli dell'accelerometro che è alimentato a 3,3volt. Il LED D2 fornirà come vedremo più avanti indicazioni sullo stato di funzionamento, mentre il pulsante SW1 ci servirà per avviare la procedura di taratura. Il trimmer R12 viene utilizzato per impostare rapidamente un parametro di funzionamento.

Il sensore infrarosso U3 pur essendo stato inserito e montato, al momento non viene utilizzato ma in un prossimo sviluppo potrebbe servire per un comando da remoto. Da ultimo facciamo notare come il valore del quarzo (4MHz) sia corretto; nel nostro caso il clock a 48MHz viene ottenuto partendo dalla frequenza di 4MHz ed innalzandola tramite la PLL interno al microcontrollore.

COSTRUZIONE MECCANICA E ASSEMBLAGGIO

Nelle foto che illustrano l'articolo potete notare la semplicità costruttiva del nostro robot che prevede una struttura verticale il cui unico scopo è quello di posizionare la batteria in alto. Questo fatto ci permette di raggiungere alcuni obiettivi. Il primo riguarda il baricentro che così facendo non può che risultare molto alto; in questo modo anche i lettori più scettici non potranno dubitare dell'instabilità della struttura sospettando l'impiego di chissà quali trucchi per far rimanere in verticale il robot. In secondo luogo, pur ricordando quella di un pendolo, la nostra struttura non ha un perno fisso dal quale può oscillare bensì sono le ruote che si spostano mantenendo il punto di appoggio sempre perfettamente verticale al centro di gravità. Avere molto peso in cima ed una struttura leggera in basso è un



Schema a blocchi del nostro robot. La tensione di alimentazione della scheda di controllo viene prelevata dall'uscita a 5 V del driver MD23.

vantaggio perchè si crea un'elevata inerzia in alto lasciando più agilità alla zona bassa dove agiscono i motori. Così, per un gioco di masse, risulterà che la parte alta del robot rimane più ferma, aumentando il senso di stabilità. Una prova simile potete farla anche voi utilizzando sempre lo stesso ombrello ed aggiungendo del peso prima in punta e poi in basso (quella a contatto del dito): capirete immediatamente i concetti sopra esposti e noterete subito i miglioramenti nella stabilità dell'ombrello. Un ragionamento simile viene fatto dai giocolieri con i loro attrezzi nei quali la distribuzione delle masse riveste grande importanza. L'intero progetto è stato dimensionato e realizzato tenendo in debita considerazione tutti gli aspetti riguardanti i pesi e le dimensioni. Nulla vieta di realizzare strutture diverse ma è necessario adattare alcuni parametri nel firmware del microcontrollore. Oltre alle immagini pubblicate in queste pagine, sul sito della rivista è disponibile un filmato dal quale è possibile ricavare molteplici informazioni relative al funzionamento del robot.

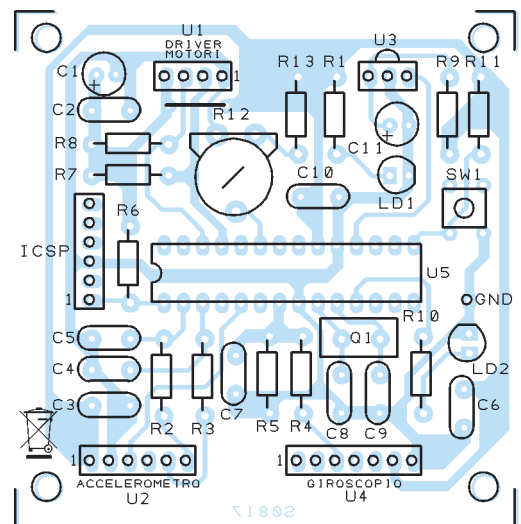
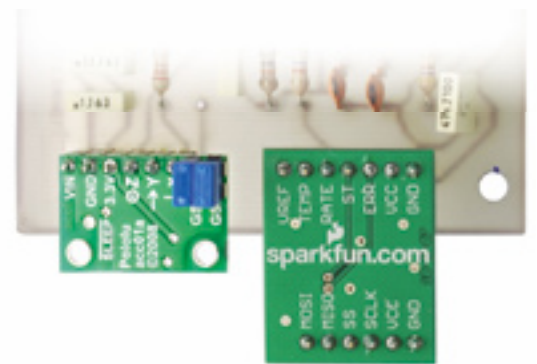
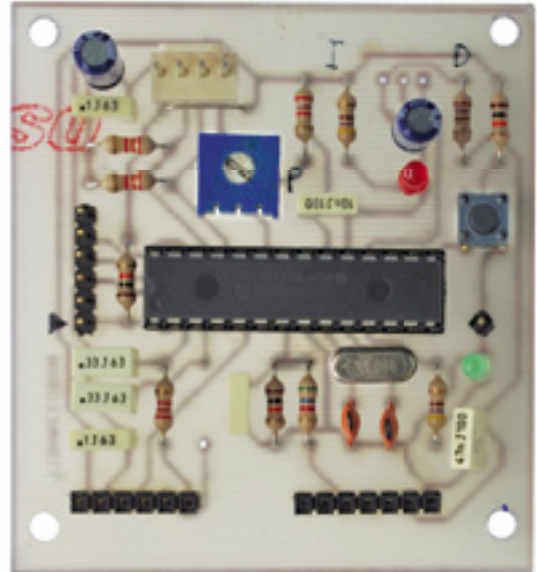
Per ultimo riportiamo anche i dati della batteria utilizzata in questo prototipo: si tratta di una batteria ai polimeri di litio con 3 celle con una capacità di 2100 mAh e con un peso di circa 150 grammi. Questo tipo di batteria dispone di un elevato rapporto potenza/peso ed è in grado di fornire elevati spunti di corrente, cosa molto importante in questo progetto in quanto i motori sono costantemente soggetti a picchi di assorbimento dovuti a repentine accelerazioni e cambi di direzione.

IL FIRMWARE

La parte decisamente più ostica

Elenco Componenti:

- R1: 470 ohm
 - R2: 1 kohm
 - R3: 1 kohm
 - R4: 5,6 kohm
 - R5: 10 kohm
 - R6: 10 kohm
 - R7: 2,2 kohm
 - R8: 2,2kohm
 - R9: 220 ohm
 - R10: 470 ohm
 - R11: 10 kohm
 - R12: trimmer MO 10 kohm
 - R13: 5,6 kohm
 - C1: 10 µF 63 VL elettrolitico
 - C2: 100 nF 63 VL poliestere
 - C3: 100 nF 63 VL poliestere
 - C4: 330 nF 63 VL poliestere
 - C5: 330 nF 63 VL poliestere
 - C6: 47 nF 100 VL poliestere
 - C7: 220 nF 63 VI poliestere
 - C8: 22 pF ceramico
 - C9: 22 pF ceramico
 - C10: 10 nF 100 VL poliestere
 - C11: 10 µF 63 VL elettrolitico
 - LD1: Led 3 mm rosso
 - LD2: Led 3 mm verde
 - SW1: Microswitch
 - U1: Driver motori 12Vdc/1A (MD23)
 - U2: Accelerometro (MMA7260QT)
 - U3: PNA4602
 - U4: Giroscopio (MLX90609-300)
 - U5: PIC18F2550-I/SP (MF817)
 - Q1: Quarzo 4 MHz
- Varie:
- Zoccolo 14+14
 - Strip maschio 4 poli
 - Strip maschio 6 poli
 - Strip femmina 6 poli
 - Strip femmina 7 poli
 - Circuito stampato



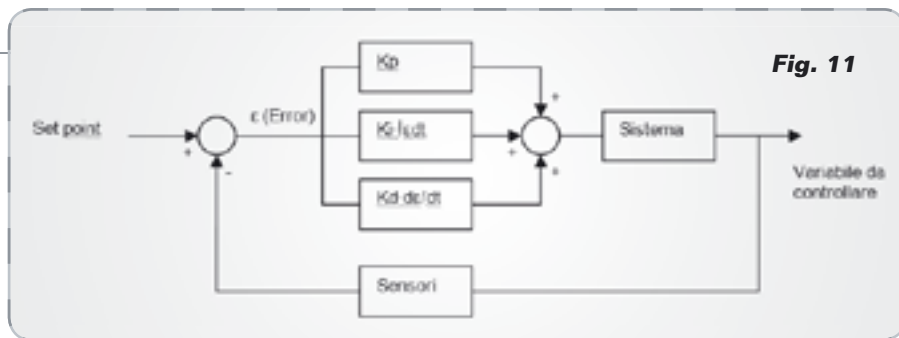
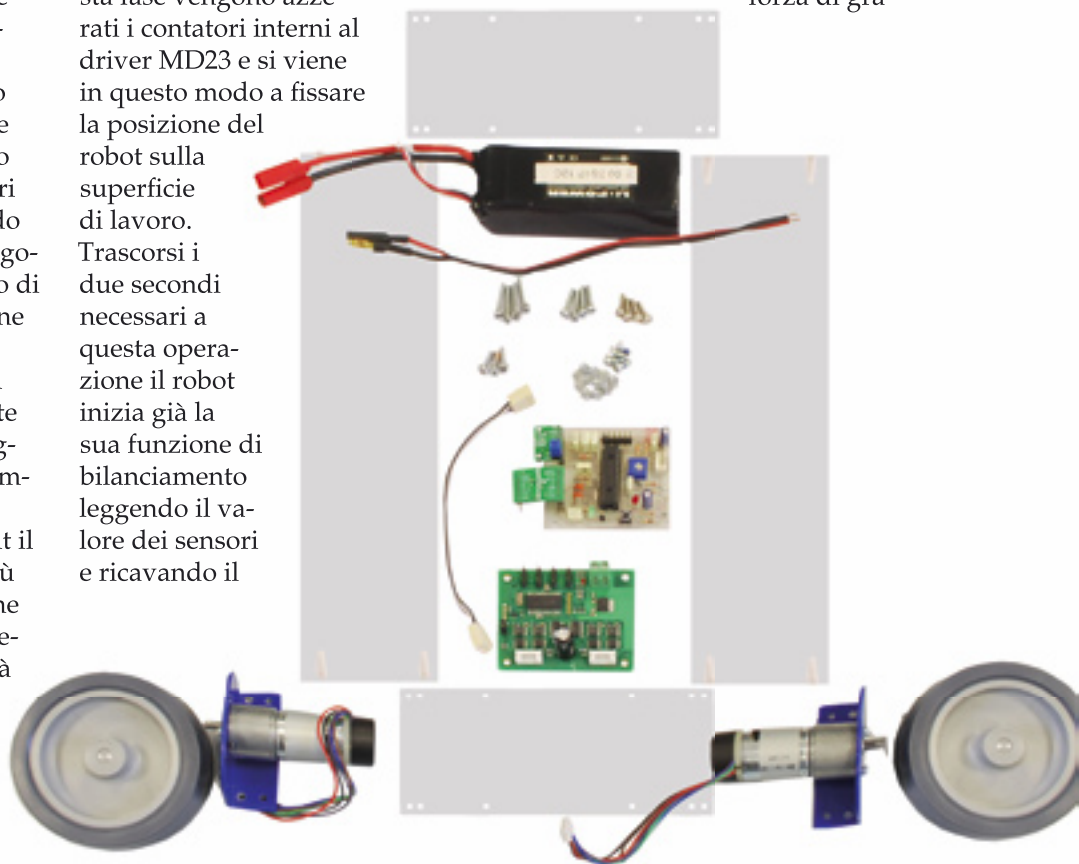


Fig. 11

in questo progetto è proprio lo sviluppo del firmware per il microcontrollore, dall'equazione per la fusione della lettura dei due sensori sino al controllo vero e proprio dei motori la strada è lunga e piena di insidie. La prima operazione svolta dal firmware dopo l'alimentazione è quella di impostare opportunamente i registri interni del PIC, successivamente vengono dichiarate le variabili utilizzate e a seguire viene programmato il driver di comando dei motori MD23. Viene impostato il modo 2, disabilitata la funzione di regolazione automatica del numero di giri e viene posta l'accelerazione a 5 e velocità iniziale zero. A questo punto viene avviato un loop di attesa premuta pulsante durante il quale il LED lampeggia rapidamente, allo stesso tempo viene testato il livello della batteria e se inferiore a 11,0volt il lampeggio del LED diventa più lento. In nessun altro caso viene testata la batteria per non impegnare ulteriormente la CPU già occupata a svolgere calcoli più importanti. Premendo il tasto il LED si spegne e si hanno a disposizione quattro secondi per posizionare il robot esattamente sulla verticale in posizione bilanciata, a seguire il LED si accende e si attiva la fase di taratura dell'offset dei due sensori. Per una lettura molto precisa vengono fatte diverse misure e poi ricavata la media dei valori. In questo modo si ottiene un valore più preciso e con risoluzione maggiore rispetto le reali capacità dei convertitori. Configurando i convertitori

interni al microcontrollore per lavorare a 10bit con la media aritmetica si ottiene una risoluzione di circa 12bit. Questo valore sarà sottratto di volta in volta dalla misura dei sensori per ottenere un valore relativo alla variazione della posizione del robot. In questa fase vengono azzerati i contatori interni al driver MD23 e si viene in questo modo a fissare la posizione del robot sulla superficie di lavoro. Trascorsi i due secondi necessari a questa operazione il robot inizia già la sua funzione di bilanciamento leggendo il valore dei sensori e ricavando il



valore dell'angolo di inclinazione necessaria per stabilire il corretto controllo dei motori. Anche in questo caso la lettura dei sensori viene fatta leggendo rapidamente più volte il loro valore ed eseguendo la media aritmetica. Questo è necessario perché l'inclinazione con cui facciamo lavorare il sensore è di pochissimi gradi e la

deviazione del segnale analogico dei sensori è di pochi millivolt. Una semplice lettura dei sensori non fornirebbe un valore adeguatamente preciso. L'intero ciclo di lettura e controllo viene eseguito 50 volte al secondo consentendo una sufficiente rapidità di intervento. A questo punto, viene implementata la fusione della lettura dei due sensori ottenendo la stima dell'angolo di inclinazione del robot rispetto la verticale rappresentata dalla direzione della forza di gra-

fornisce indicazione sullo spazio percorso dal robot ovvero dalla sua distanza dal punto in cui è avvenuta la taratura. Il valore del contatore è un numero con segno a 32 bit con il quale possiamo ricavare non solo lo spazio percorso ma anche la direzione rispetto al punto di partenza. Non ci resta che determinare quale valore di velocità dovrà essere impartito al motore. Ogni 20 msec andiamo ad aggiornare il registro *Speed1* del driver MD23 con la nuova velocità che dovranno avere i motori. In *mode2*, infatti, scrivendo la velocità sul registro *Speed1* dell'MD23, vengono impostate contemporaneamente le velocità dei due motori. La differenza di velocità impostata tra una scrittura e l'altra rappresenta anche l'accelerazione impartita ai motori, valore essenziale, in quanto rappresenta la *forza* con la quale contrastiamo la caduta del robot. Per eseguire questa operazione dobbiamo ricorrere alla teoria dei controlli che prevede proprio la gestione di una variabile di uscita in funzione del valore letto da un sensore che la misura. Lo schema di principio è quello descritto in Figura 11.

Il *set point* rappresenta per il nostro sistema l'angolo che vogliamo mantenga il robot, ovvero zero, in quanto vogliamo rimanga perfettamente in verticale. Con il termine *Sistema* rappresentiamo l'insieme delle parti costituenti il robot e cioè il driver, i motori, le ruote, la struttura meccanica con i vari pesi in gioco. Con il box *Sensori* indichiamo invece il comportamento dei sensori o meglio della variabile "angolo" ricavata dalla lettura dei sensori. La variabile di uscita nel nostro caso è l'angolo di inclinazione del robot rispetto la verticale. Il sistema di controllo composto dai tre blocchi a

PUMA, il SEGWAY da strada

Il principio di funzionamento è quello del primo Segway, ma questa volta le ambizioni sono decisamente maggiori. Grazie all'alleanza con General Motors che prevede di iniziare la commercializzazione nel 2012, è stato presentato pochi giorni fa il prototipo del PUMA (Personal Urban Mobility and Accessibility), un veicolo elettrico a metà strada tra una carrozzella per disabili e una city-car tagliata a metà. Un veicolo studiato per la città, in grado di trasportare due persone ad una velocità massima di poco meno di 60 km/h e con un'autonomia di 60 chilometri. Anche in questo caso sofisticate soluzioni elettromeccaniche assistite da un giroscopio e da altri sensori consentono al veicolo di restare in equilibrio sulle due ruote in qualsiasi condizione ed a qualsiasi velocità. Un mezzo ideale per il traffico cittadino con un costo di esercizio

destra dell'errore rappresentano il controllo del sistema che in modo abbreviato viene indicato con la sigla PID acronimo di *Proporzionale, Integrativo, Derivativo*, in relazione alla funzione di ogni singolo blocco. Come si vede dallo schema, il sistema è autocompensante, in quanto se la variabile di uscita

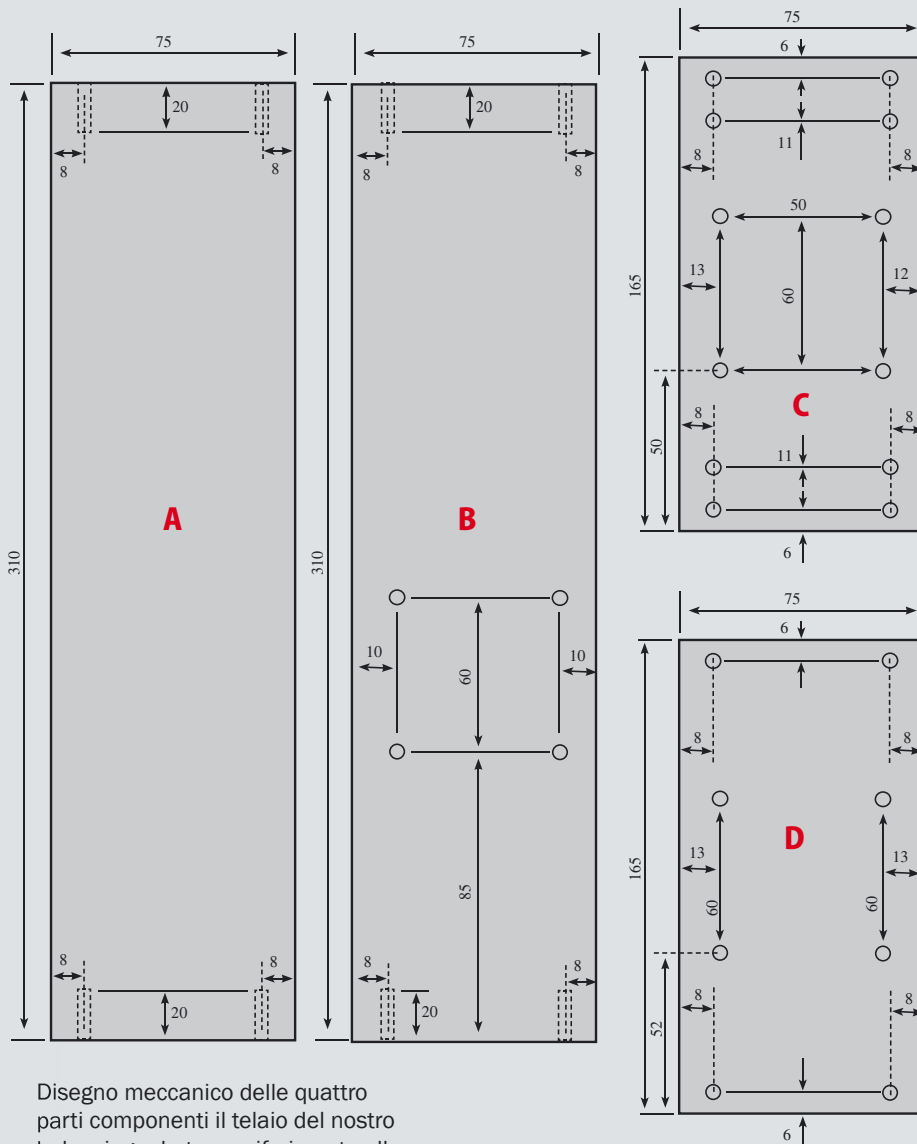
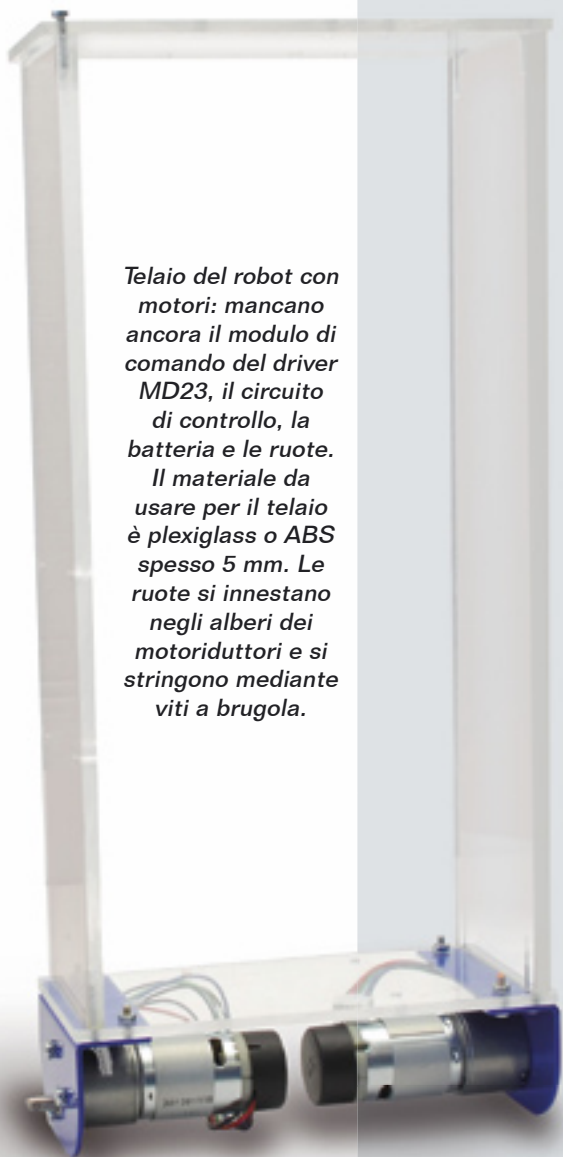
irrisorio. Ed a guardare i video che circolano in Internet sembra proprio che questo trabiccolo se la cavi piuttosto bene nel traffico cittadino, agile quanto una moto ma con un tetto sulla testa e con due comodi sedili. Certo, fare un frontale con questo mezzo non deve essere una cosa piacevole, così come affrontare strade sconnesse, discese o curve pericolose; anche il costo (non ancora comunicato) non sarà dei più abbordabili, visto quanto costa il Segway tradizionale. Probabilmente in questo momento di grave crisi della GM - che è ad un passo dalla bancarotta - prodotti del genere distolgono l'attenzione, almeno per un po' di tempo, da problemi e scelte molto più importanti.

dovesse aumentare, all'ingresso del blocco *Sistema* arriverebbe un segnale minore (il contrario se la variabile in uscita dovesse diminuire). In realtà nel nostro caso il tutto è un pochino più complesso in quanto misuriamo anche la distanza percorsa dal robot tramite gli encoder e dobbiamo tenerne conto quando facciamo il



L'assemblaggio del robot

Telaio del robot con motori: mancano ancora il modulo di comando del driver MD23, il circuito di controllo, la batteria e le ruote. Il materiale da usare per il telaio è plexiglass o ABS spesso 5 mm. Le ruote si innestano negli alberi dei motoriduttori e si stringono mediante viti a brugola.



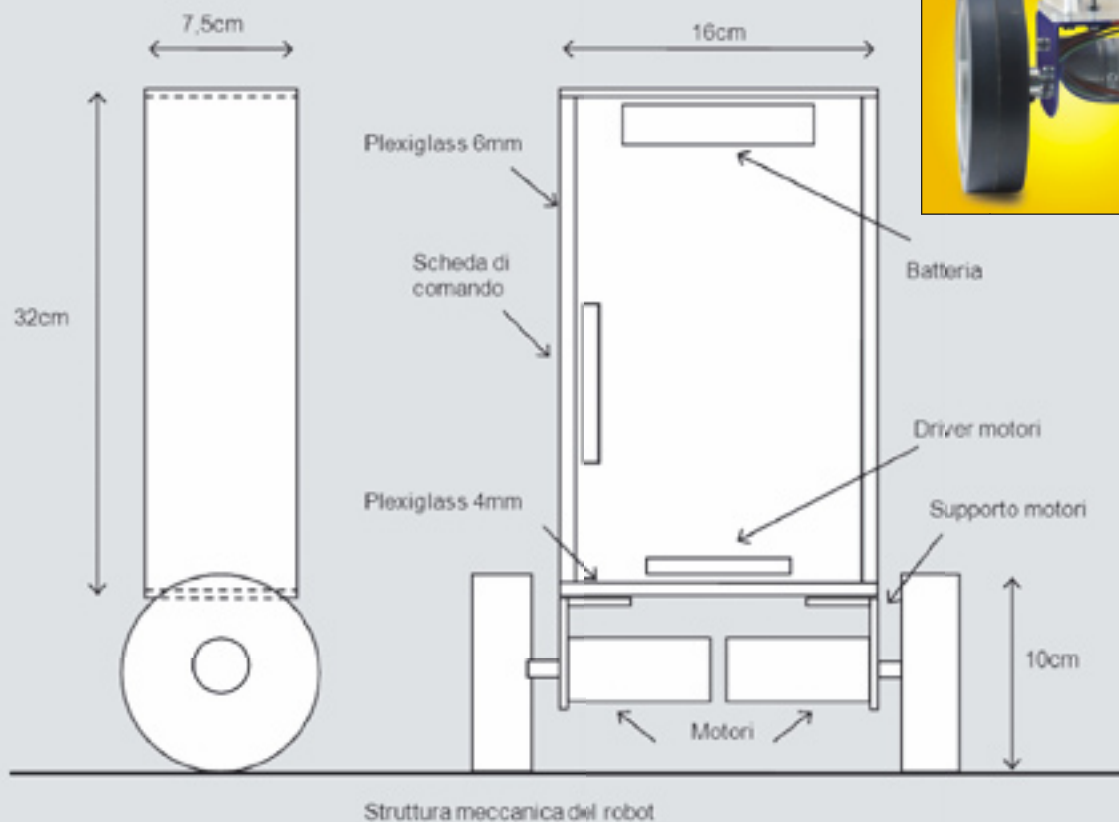
Disegno meccanico delle quattro parti componenti il telaio del nostro balancing-robot: con riferimento alla figura qui a sinistra, **A** è il fianco destro (ossia quello cui non è fissato alcunché) **B** è il fianco sinistro (prevede la foratura per il fissaggio del

controller MD23, **C** è la base (ossia la piastra che supporta i fianchi e la scheda di controllo e alla quale si

controllo del sistema. Lo schema proposto, inoltre, è valido per il campionamento di un segnale *continuo* nel tempo; tuttavia nel nostro caso, lavorando ad intervalli di 20 msec, ci troviamo nella fattispecie di un segnale intrinsecamente *discreto* nel tempo, per il quale è necessario ragionare in modo leggermente differente. Senza entrare troppo nelle complesse implicazioni matematiche, soffermiamoci sulle tre

costanti denominate **K_p**, **K_i** e **K_d** che risultano fondamentali per la nostra applicazione e saranno loro che opportunamente tarate permetteranno la stabilità del sistema. La prima costante **K_p**, determina la reazione del sistema quando è soggetto ad una determinata inclinazione. La seconda costante **K_i** determina la reazione del sistema alle variazioni di inclinazione perchè moltiplica il termine $d\epsilon/dt$ che rappresenta

la variazione dell'angolo a cui è sottoposto il robot (essendo il set poi uguale a zero). La terza costante **K_i** determina la reazione del sistema nel lungo termine ovvero nel caso in cui il robot rimanga inclinato per parecchio tempo (stiamo comunque parlando di millisecondi). Una quarta costante denominata **K_{Pp}** (dove la seconda *P* sta per *position*) determina la reazione del sistema ad uno spostamento



fissano i gruppi motore) e, infine, **D** è il “tetto”, ossia la parte che chiude superiormente e raccorda i fianchi. Tutte le quote sono espresse in millimetri. Le parti tratteggiate sui lati corti dei fianchi rappresentano i fori con filettatura metrica da 3 mm e servono per avvitare i pezzi **C** e **D** su **A** e **B**, cosa che si fa usando delle viti (3MA) da 25 mm di lunghezza. I fori (due per parte) vanno realizzati dai lati stretti e tenendo in piedi i fianchi, cercando di tenere questi ultimi paralleli alla punta

del trapano, così che vengano i più dritti possibile; la punta da usare è da 2,5 mm, perché poi a 3 mm si arriva con una filiera maschio. Qui sopra, il disegno d'insieme del robot e le principali fasi del montaggio: per prima cosa si fissano i gruppi motore alla base mediante le viti più interne, poi, nei fori esterni si mettono le viti che terranno insieme base e fianchi. A questo punto si avvita anche la parte superiore e sugli alberi dei motoriduttori si stringono le ruote.

rispetto alla posizione di appoggio iniziale. Anche in questo caso sono disponibili le costanti **K_{Pi}** e **K_{Pd}** relative all'integrale ed alla derivata dello spostamento ma di fatto sono state poste a zero. In teoria sarebbe possibile determinare a tavolino queste variabili per un funzionamento ottimale, ma ciò comporterebbe la conoscenza del comportamento di ogni singolo componente.

COLLAUDO E FUNZIONAMENTO
In considerazione della delicatezza del progetto e della complessità del firmware, la realizzazione di questo robot è consigliata ai lettori con una buona esperienza in questo campo; seguendo scrupolosamente le nostre indicazioni di montaggio, specie per quanto riguarda la struttura meccanica ed i componenti elettromeccanici, sicuramente il risultato finale sarà raggiunto. Nulla vieta, tut-

tavia, specie se possedete l'esperienza necessaria, di utilizzare una propria struttura meccanica. In ogni caso dovrete tarare attentamente le costanti **K_p**, **K_i**, **K_p** e **K_p** per ottenere la migliore stabilità; a questo proposito forniremo tra breve tutte le indicazioni. Agendo direttamente sul firmware avrete la possibilità di modificare per tentativi successivi le variabili fino ad ottenere il valore ottimale. Il firmware



per il MATERIALE

Tutti i componenti utilizzati per realizzare questo robot sono facilmente reperibili in commercio. Il master del circuito stampato ed il firmware per programmare il microcontrollore sono scaricabili gratuitamente dal sito della rivista. Il micro è anche disponibile già programmato al prezzo di 14,00 Euro (cod. MF817). L'accelerometro (MMA7260QT) costa 28,00 Euro, il giroscopio (MLX90609-300) 75,00 Euro ed il set di trazione (RD01) 220,00 Euro. Tutti i prezzi includono l'IVA.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11,
21013 Gallarate (VA)

Tel: 0331-799775 • Fax: 0331-792287

<http://www.futurashop.it>

determina anche, durante la fase di taratura, il valore di offset dei sensori, cosa molto importante per determinare l'esatto asse verticale. Una precisa impostazione è possibile solo se il robot è posizionato esattamente sulla verticale ovvero si trova in una posizione di bilanciatura forzata anche assistita manualmente. Dopo aver acceso il robot la procedura di taratura prevede le seguenti fasi:

1. Robot acceso, LED lampeggiante veloce.
2. Appoggiare il robot nella superficie di lavoro.
3. Premere il pulsante, il LED si spegne.
4. Entro quattro secondi posizionare il robot esattamente in verticale.
5. Il LED si accende a luce fissa, fase di lettura offset sensori.
6. Dopo due secondi il LED si spegne: la taratura è ultimata.
7. Rilasciare delicatamente il robot.
8. Il robot rimane bilanciato.

Ovviamente se durante la fase di taratura dell'offset il robot non fosse perfettamente in posizione verticale, questa inclinazione sarebbe mantenuta anche di

seguito con lo svantaggio di vedere il robot muoversi nel verso dell'inclinazione senza mantenere la posizione originale. La determinazione delle costanti di controllo viene fatta sperimentalmente agendo direttamente sui valori scritti nel firmware, uno alla volta, secondo il seguente procedimento:

1. Porre tutte le variabili **Kp**, **Ki**, **Kd** a zero.
2. Il robot cade e la base non si muove.
3. Incrementare il valore di **Kp**.
4. Ripetere i passi 3-4 finché la base compie piccole oscillazioni ed il robot rimane in verticale anche per pochi secondi. Se **Kp** è troppo piccolo la base insegue la cima del robot mentre l'inclinazione aumenta mentre se **Kp** è troppo grande le ruote tremano e la base vibra velocemente.
5. Incrementare **Ki** finché il robot rimane bilanciato per alcuni secondi con una oscillazione costante. Quando si aumenta il valore di **Ki** la base non accelera rapidamente se l'angolo cambia di segno (e viceversa). Il robot tende ad oscillare lentamente attorno alla condizione di bilanciamento che è conosciuta come *condizione di overshoot*.
6. Incrementare **Kd** allo stesso modo di **Kp** e **Ki** finché la condizione di *overshoot* sparisce ed il pendolo rimane bilanciato.
7. Quando l'*overshoot* è sparito il pendolo è bilanciato.

La condizione ideale di perfetta bilanciatura potrebbe non essere raggiunta, ottenendo comunque una sufficiente stabilizzazione che mantenga il robot in verticale. Successivamente si procede ad incrementare la variabile **KPp** ed eventualmente **KPi** e **KPd**, sino a quando il robot rimane posi-

zionato nell'intorno del punto di taratura della base di appoggio. Ci rendiamo conto che i concetti espressi non sono dei più facili e pur avendo semplificato al massimo la trattazione possiamo immaginare le perplessità di qualche lettore. Nello sviluppo del nostro prototipo molti sono stati i fallimenti e le difficoltà incontrate soprattutto durante le prime prove: nulla però ci ha scoraggiati, anzi, da tutto ciò siamo stati spronati ad approfondire l'argomento. In ogni caso la soddisfazione di vedere il robot in verticale ha ricompensato i nostri sforzi. A questo punto più di un lettore si chiederà quale possa essere lo scopo pratico di questo progetto al di là della mera didattica: nell'immediato sicuramente nessuno ma l'esempio del Segway - che sfrutta gli stessi concetti - dovrebbe fare riflettere quanti sono sempre alla ricerca di un'applicazione pratica. ■

GUARDA IL NOSTRO ROBOT SU YOUTUBE



Se volete vedere il nostro robot correre qua e là, persino su un tavolo o su un piano inclinato, potete guardare il filmato pubblicato su YouTube all'indirizzo http://www.youtube.com/watch?v=ngv_De2ISdg. Sullo stesso sito troverete i filmati degli altri robot presentati in passato sulle pagine della nostra rivista.