Gli ingranaggi del ruotismo anteriore

Le due serie di ingranaggi della pinza, quella anteriore e quella posteriore, sono di fatto dei ruotismi, cioè sistemi per la trasmissione del moto costituiti da più di due elementi (di solito ruote). Il ruotismo anteriore, in particolare, verrà costituito da un primo rocchetto (vale a dire una ruota dentata piccola) che trasmette la rotazione del motore a spazzola a un ingranaggio formato da due ruote dentate di pari diametro; inoltre, grazie a un secondo rocchetto, il ruotismo verrà collegato alle slitte a cremagliera dei becchi. Queste ultime, di fatto, trasformeranno il moto rotatorio di ruote e rocchetti nel moto rettilineo necessario per aprire e chiudere la pinza. Appare allora evidente che il ruotismo dipende, in ultima analisi, dal motore a spazzola e dalla rotazione che esso impone alla vite senza fine che, di conseguenza, risulta essere il primo degli elementi del ruotismo stesso.

Gli ingranaggi anteriori trasmetteranno ai becchi della pinza la rotazione del motore, trasformandola in moto rettilineo

CONTROLLO DEI MOTORI

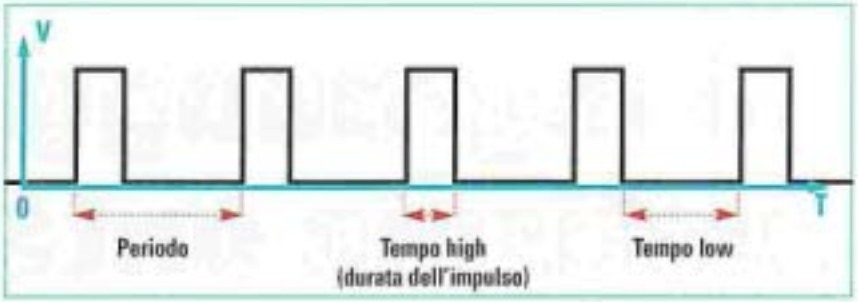
Applicando una tensione continua ai poli di un motore a spazzola in corrente continua (come quello che azionerà il ruotismo del blocco anteriore del manipolatore), esso produce una rotazione della vite senza fine a tutta velocità: non è cioè possibile esercitare alcun tipo di controllo su di esso, salvo dare o togliere alimentazione.

In realtà, è possibile controllare anche un motore a spazzola con la cosiddetta tecnica PWM (acronimo di *Pulse With Modulation*, ossia 'Modulazione A Impulsi'), basata sulla gestione di opportuni impulsi di pilotaggio, modulati e variabili. Nel caso particolare del microcontrollore BASIC STAMP 2, però, per quanto funzionale ed efficace, esso non può permettersi i tempi necessari per il controllo dei motori a spazzola in PWM: se lo facesse, infatti, non potrebbe poi svolgere tutte le altre funzioni richiestegli (ad esempio, gestire gli input e gli output, oppure eseguire subroutine) che, invece, sono prioritarie per il complessivo funzionamento del robot. Per controllare i motori a spazzola in PWM, allora, occorrerebbe affiancare al microcontrollore un altro, più sofisticato circuito di controllo, dedicato unicamente a tale funzione.

Sopra. Le ruote dentate 1, i rocchetti 2 e i rispettivi perni 3, allegati a questo fascicolo, andranno a costituire il ruotismo del blocco anteriore della pinza.

SERVOMOTORI

È quanto accade, ad esempio, con i servomotori, cui però, nel caso del tuo robot, è stata assegnata la gestione della locomozione. **Un servomotore, infatti, non è che un motore dotato di un circuito di autocontrollo** che, in base alle caratteristiche dell'impulso che riceve, genera al suo interno un segnale PWM (cioè, come vedremo, un treno di impulsi) utile per controllare il proprio movimento in velocità e con buona precisione (tramite un apposito driver, anch'esso interno). I servomotori standard (impiegati per esempio nel radiomodellismo) hanno un meccanismo di controreazione che, in base all'impulso che riceve, consente unicamente il posizionamento desiderato, purché inferiore a 360 gradi (tipicamente fino a 90°, 180° oppure 270°). Per controllare le ruote e, dunque, la locomozione del robot, i servomotori Parallax, sono stati invece modificati in modo tale che, dato un treno di impulsi, la ruota possa compiere rotazioni di 360°, cioè complete. **Comunque, per pilotare un qualsiasi servomotore è necessario anzitutto alimentarlo**: per quanto riguarda il robot, bisogna collegarlo sia a Vss (cioè la massa, pari a 0 volt) sia a Vin (cioè le batterie, per un massimo di 6 volt in corrente continua). **Inoltre, sempre in ingresso, il servomotore richiede un impulso o, meglio, una PWM, cioè una serie di impulsi** (il cosiddetto *pulse train*, ossia 'treno di impulsi'), che viene poi 'tradotto' in un corrispondente moto rotatorio. Come vedremo nelle prossime pagine, l'input relativo al treno di impulsi viene definito in sede di programmazione e si ottiene con un particolare comando PBASIC (**pulsout**).



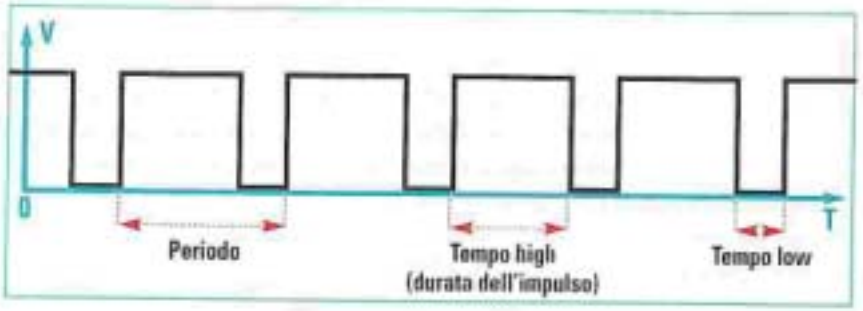
Sopra. Forma d'onda di un treno di impulsi, riferito a due assi cartesiani (l'ascissa corrisponde al tempo T; l'ordinata alla tensione V). La somma del tempo high (relativo a una tensione alta) e di quello low (tensione bassa) corrisponde al cosiddetto periodo.

PERIODO E DUTY CYCLE

Un impulso elettrico non è che una brusca e momentanea variazione della tensione di un segnale da uno stato all'altro (alto-basso o viceversa): i parametri da cui dipende, dunque, sono due, cioè la durata dell'intervallo di tempo in cui il segnale permane a un certo livello di tensione e il valore di quest'ultima. Lo stesso vale per un treno di impulsi (illustrato sopra, in questa e nella pagina accanto), nient'altro che una successione ciclica, ossia periodica, di impulsi. **In pratica, in un treno di impulsi si alternano valori di tensione alti (*high*) e bassi (*low*) a intervalli di tempo ciclici** cosicché, come per ogni altro segnale periodico, tutte le informazioni sono di fatto contenute, e possono essere osservate, in un unico particolare intervallo di tempo (il periodo), tipicamente scelto a partire dall'istante in cui insorge un dato impulso, fino a quello che precede

Nella tabella. Confronto tra alcune caratteristiche e modalità del moto prodotto da un motore a spazzola e da un servomotore. La velocità di rotazione nei due sensi, orario e antiorario, fissata per i motori a spazzola, nei servomotori può essere controllata variando il valore di *pulsout* rispetto a quello equivalente all'arresto dei motori (tarato a 750).

DESCRIZIONI	MOTORE A SPAZZOLA	SERVOMOTORE
Alimentazione	min 3 volt - max 6 volt	min 5 volt - max 6 Volt
Pilotaggio	diretto	a impulsi (min 500 - max 1000)
Arresto	cavi scollegati oppure poli allo stesso potenziale	con impulso di 1.5 ms (pulsout 750)
Rotazione oraria	cavo rosso = polo negativo cavo nero = polo positivo	con impulso da 1 a 1.5 ms (pulsout tra 500 e 750)
Rotazione antioraria	cavo rosso = polo positivo cavo nero = polo negativo	con impulso da 1.5 a 2 ms (pulsout tra 750 e 1000)



Sopra. **Un treno di impulsi con un diverso duty cycle (rapporto tra tempo high e periodo) rispetto a quello illustrato a pag. 112 (qui il tempo high è maggiore).**

immediatamente l'insorgere dell'impulso successivo (come illustrato in alto, in questa pagina e nella precedente). **Trascorso un periodo, allora, il segnale si ripete in modo del tutto identico.** Il periodo di un treno di impulsi, in particolare, è la somma di un primo intervallo di tempo (detto 'tempo high'), in cui il segnale si porta e si mantiene al livello di tensione alto, e di un secondo intervallo ('tempo low'), in cui il segnale si riporta al livello di tensione basso. Il rapporto tra il tempo high e il periodo, invece, viene detto *duty cycle* ('ciclo di lavoro') e può essere espresso come percentuale, considerando il periodo come il 100%. Nell'illustrazione a pag. 112, ad esempio, il duty cycle può essere calcolato come 30% (il tempo high è circa un terzo del low); il treno di impulsi illustrato qui sopra, viceversa, ha un duty cycle approssimabile al 70% del periodo. In pratica, quindi, il duty cycle rappresenta la percentuale di tempo in cui il segnale rimane alto: è a tale valore di tensione, infatti, che il segnale è 'utile' come impulso, ossia effettua il lavoro richiestogli (ad esempio, il controllo dei servomotori).

TEMPI ALTI E BASSI

Il tempo high, che viene definito anche 'durata' dell'impulso, è il principale parametro utile per governare i servomotori. Esso determina infatti il senso di rotazione (orario o antiorario) imposto alle ruote del robot. In particolare, date le specifiche costruttive Parallax, i valori che andremo a utilizzare sono anzitutto una tensione low pari alla massa (cioè Vss=0 V) e una tensione high fornita dal microcontrollore (pari cioè a 5 V). Per quanto riguarda invece gli intervalli di tempo, **il tempo high dovrà essere compreso fra 1 e 2 millisecondi (ms), il tempo low, invece, andrà da 10 a 40 ms.** Dunque il periodo dell'impulso (somma dei due tempi) sarà compreso fra 11 ms (1+10) e 42 ms (2+40). Indagato il concetto di treno di impulsi da un punto di vista elettronico, vediamo ora come generarlo e gestirlo a livello di programmazione, tramite i comandi **pulsout** e **pause**.

PARAMETRI DI RIFERIMENTO E UNITÀ DI MISURA

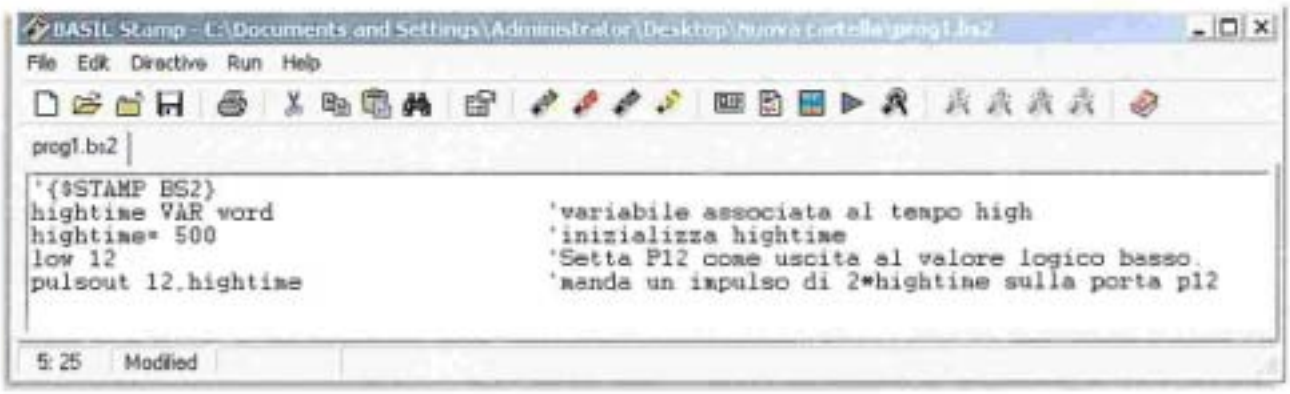
Come abbiamo più volte avuto modo di constatare, alcuni dei parametri necessari nella fase di programmazione del robot dipendono unicamente dalle specifiche costruttive di uno o più componenti. In particolare, per quanto riguarda la programmazione dei servomotori, tali parametri dipendono dalle scelte della Parallax. Un primo parametro che ti sarà utile nelle prossime pagine è di tipo temporale: si tratta di quella sorta di unità di misura corrispondente a 2 microsecondi (µs), già incontrata a proposito della costante di tempo calcolata dal comando rctime (a pag. 108). Anche il comando pulsout che, come vedremo, gestisce il tempo high è tarato su questo parametro: pulsout 1, cioè, corrisponde a un tempo di 2 µs, pulsout 2 a un tempo di 4 µs (2*2 µs) ecc. Tali valori espressi da pulsout, di fatto, non sono altro che l'equivalente della durata dell'impulso da generare, 'tradotto' in un'altra unità di misura.

Così, ad esempio, per generare un impulso di 1.5 ms (che equivale a 1500 µs), è necessario assegnare a pulsout un valore pari a 750 (1500 µs diviso 2 µs). Poiché, come si è detto, il tempo high deve essere compreso tra 1 e 2 ms, appare chiaro che 1.5 ms non è un valore qualsiasi, bensì l'intermedio dell'intervallo considerato. Di conseguenza, anche pulsout 750 risulterà essere un valore tipico: in particolare, è la soglia di arresto dei servomotori. Per valori di pulsout compresi tra 500 (1 ms=1000 µs diviso 2 µs) e 750, il servomotore ruoterà in senso orario, mentre per valori compresi tra 750 e 1000 (2 ms=2000 µs diviso 2 µs), ruoterà in senso antiorario. Per quanto riguarda invece il tempo low (generalmente compreso tra 10 e 40 ms), un parametro utile per la programmazione è quello pari a 20 ms. A differenza di pulsout, però, il comando relativo al tempo low (pause) ha per unità di misura il millisecondo (ms) e, dunque, mantiene inalterato il valore della cifra indicata.

Le fasi di programmazione

Vediamo ora in dettaglio il comando pulsout per la gestione della durata dell'impulso che si vuole generare per controllare i servomotori.

Mandando in esecuzione il listato qui sotto, potrai osservare che la ruota corrispondente al servomotore collegato alla porta P12 ruoterà in senso orario a una certa velocità. Come si diceva nel box di pag. 113, questo dipende dalla durata dell'impulso relativo al tempo high che, nel programma, si traduce nel valore (500) della variabile (var o VAR, che è lo stesso: maiuscole, minuscole e spazi, infatti, non incidono sulla programmazione) di tipo word, etichettata con *hightime*. Cambiando tale valore, cambia l'angolo di rotazione della ruota e, oltre la soglia di 750, il senso. In particolare, per valori di *hightime* compresi tra 500 e 750, la rotazione è oraria e l'angolo diminuisce via via che ci si avvicina a 750. A 750, il motore si arresta, mentre per valori di *hightime* compresi tra 1000 e 750 la rotazione sarà antioraria e l'angolo aumenta via via che ci si avvicina a 1000.

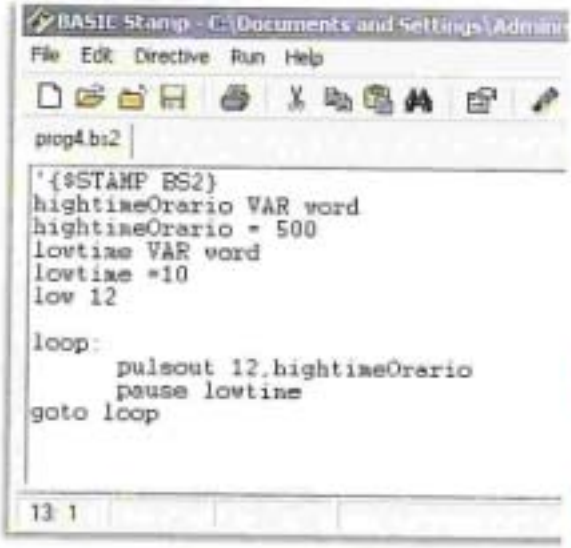


Il listato qui sotto esegue automaticamente una prova di quanto detto: grazie a un ciclo for, infatti, incrementa *hightime* di 50 (cioè di 50*2 µs=100 µs) a ogni iterazione, verificandone il comportamento a partire dal valore 500, fino a 1000. Vediamo ora il listato in dettaglio. Dichiarata la variabile *hightime* di tipo word, come già nel listato precedente, compare il comando low 12, tramite il quale la porta P12 viene impostata come output e il suo valore logico è settato basso. Segue quindi il comando pulsout (sotto è inserito nel ciclo for), la cui sintassi prevede che al comando segua la *porta* interessata (12), una virgola (,) e, quindi, il *tempo* memorizzato in *hightime*. Come si diceva, questo *tempo* corrisponde al valore indicato moltiplicato per 2 µs. Di fatto, allora, il comando significa: imponi sulla *porta* un impulso di durata 2 µs**tempo*. Trascorso tale intervallo di tempo, il valore di output della porta relativa viene riportato allo stato originale. Poiché, come nel nostro caso, il valore iniziale della porta è basso (low 12), l'impulso sarà sempre e comunque positivo (pulsout, cioè, porterà la P12 a un valore di tensione più alto, per poi tornare al valore basso per un certo intervallo di tempo, come vedremo tra poco).

La serie di comandi low, pulsout e pause producono una brusca e momentanea variazione di tensione (cioè un impulso). Come si diceva, perché l'impulso sia utile al robot, dovrà essere compreso tra 1 e 2 ms (cioè tra pulsout 500 e 1000). Per un impulso di durata compresa tra 1 e 1.5 ms (cioè tra pulsout 500 e 750) la rotazione sarà oraria, per impulsi di durata compresa tra 1.5 e 2 ms (750 e 1000) sarà antioraria.

Un impulso di 1.5 ms, invece, arresta il motore. Va da sé che i valori compresi tra 1 e 1.5 sono simmetrici rispetto a quelli compresi tra 1.5 e 2, laddove 1.5 è il centro di simmetria, il valore intermedio, di arresto). Perciò l'angolo di rotazione associato a un impulso di 1 ms (massimo in senso orario) è pari a quello associato all'impulso di 2 ms (massimo in senso antiorario), mentre l'angolo di rotazione associato a un impulso di 1.25 ms è simmetrico a quello associato a 1.75.

Verifichiamo quanto detto mandando in esecuzione il programma a destra che impone al motore relativo a P12 di produrre otto microrotazioni in senso orario, seguite da altre otto in senso antiorario; per tornare poi al punto di partenza. Tale prestazione si ottiene grazie a due cicli for (la cui sintassi è stata presentata nella sezione Tecno, a pag. 63) che prevedono l'incremento ciclico della variabile *counter* da 0 a 8 (poiché il passo o *step* non è stato specificato, l'incremento è quello prefissato, cioè 1). Si è assegnato alla variabile *hightimeOrario* il valore 500 e a quella *hightimeAntiorario* il valore 1000, risultato dalla sottrazione 1500-500. Tale relazione deriva dalla simmetria (750-*hightimeOrario*)+750, per cui la distanza di *hightimeOrario* dal centro di simmetria (750-*hightimeOrario*) viene sommata al valore di quest'ultimo (750), dando come risultato il corrispettivo valore simmetrico di *hightimeAntiorario*. Il fatto che *hightimeAntiorario* sia stata espressa in modo che dipenda da *hightimeOrario* (hightimeAntiorario = 1500-hightimeOrario), inoltre, migliora la modificabilità del listato: variato infatti il valore della variabile *hightimeOrario* (comunque compreso tra 500 e 750), il nuovo valore di *hightimeAntiorario* (simmetrico e, dunque, compreso tra 750 e 1000) viene aggiornato automaticamente. Variando i valori reciproci delle variabili, ovviamente, varia l'angolo di rotazione associato a ogni impulso e, dunque, lo spazio percorso dalla ruota del robot.



Generato un impulso (impressa cioè una singola microrotazione alla ruota collegata al servomotore), si tratta ora di generare un treno di impulsi che imprima una serie di microrotazioni tali da produrre un'unica rotazione complessiva. Per farlo occorre, come nel listato a sinistra, un segnale periodico (o ciclico) che si ottiene con un loop. Prima però bisogna fissare il tempo low (lowtime=10), ossia la durata dell'intervallo in cui il segnale dovrà restare basso: per un buon funzionamento dei servomotori, il tempo low dovrà essere di almeno 10 ms, ma inferiore a 40 ms (come si accennava nel box di pag. 113, un parametro ottimale è 20 ms). Dopo ogni pulsout, P12 torna al valore di tensione basso (grazie al comando pause) per l'intervallo di tempo fissato da *lowtime*. Aumentando il valore di *lowtime* (ad esempio a 40), di fatto la rotazione rallenta: maggiore è l'intervallo tra un impulso e l'altro, infatti, minore sarà la rotazione totale (somma delle singole microrotazioni). Tuttavia, la velocità di rotazione dipende anche dal valore di *hightime* (tra 500 e 750): a parità di tempo intercorso tra due microrotazioni, infatti, esse saranno più o meno brevi.

LE FASI DI PROGRAMMAZIONE

*Abbiamo visto, dunque, che le due componenti (tempo high e tempo low) che costituiscono il periodo di un segnale ciclico, come il treno di impulsi, vengono gestite dalla combinazione di due comandi diversi (rispettivamente pulsout e pause). Per attribuire loro valori coerenti, però, bisogna ricordare che, mentre il comando pause (e dunque il tempo low) ha per unità di misura il millisecondo (ms), pulsout va inteso moltiplicato per 2 microsecondi (µs). Il comando pause 10, infatti, impone un tempo low di 10 ms, mentre pulsout 500 impone un tempo high di 500*2µ=1000 µs (pari a un millisecondo, cioè 1 ms).*

Vediamo ora un programma (a destra) che gestisce la rotazione di entrambe le ruote del robot, tramite due treni di impulsi imposti ai rispettivi servomotori. Prima di procedere all'analisi del listato, è importante ricordare che i due servomotori sono stati montati in modo speculare: perché il robot proceda in una direzione, dunque, la loro rotazione dovrà essere opposta. Per andare avanti, ad esempio, il motore destro dovrà ruotare in senso orario, mentre il sinistro ruoterà in senso antiorario; viceversa, per procedere all'indietro. Se entrambi i servomotori ruotassero invece nello stesso senso, imporrebbero alle ruote una rotazione opposta e il robot girerebbe su se stesso. Sebbene per gestire la velocità di rotazione, come si è detto, sia possibile intervenire sia sul tempo high sia sul tempo low del treno di impulsi, le due operazioni non sono però del tutto analoghe. Un intervento sul tempo low, infatti, produce una variazione della frequenza degli impulsi che si registra come maggiore o minore fluidità della rotazione e si verifica che un pulsout ogni 20 ms (cioè pause 20) garantisce una buona prestazione. L'intervento sul tempo high, invece, produce una variazione di velocità molto più sensibile, utile anche per la cosiddetta 'calibrazione'. Per motivi tecnici, infatti, è probabile che due servomotori non siano identici: una piccola imperfezione, per esempio, può comportare che il moto prodotto differisca anche solo minimamente (lo stesso accade anche ai motori a spazzola; tuttavia il fatto che prevedano un'unica velocità di rotazione fa si che il problema non sia risolvibile). I servomotori, però, possono essere tarati (o calibrati), tramite opportune variazioni del tempo high rispetto allo stato di arresto (cioè il parametro intermedio ideale di 750), fino a ottenere i valori reali di arresto e, di conseguenza, un'identica velocità di rotazione. Detto questo, vediamo il listato nei dettagli. Dopo aver dichiarato le variabili (VAR) di tipo word relative al tempo high di ciascun motore (*leftHighTime* e *rightHighTime*) e a quello low valido per entrambi (*lowTime*), si procede all'assegnamento dei rispettivi valori: quelli del listato (1000, 500 e 20) faranno avanzare il robot. Perché ciò accada, però, è necessario settare le porte relative ai servomotori (low12 e low13) e inscrivere in un loop l'alternanza dei tempi high (pulsout 12, leftHighTime e pulsout 13, rightHighTime) e del tempo low (pause lowTime).

prog5.bs2

```
' {$Stamp bs2}
leftHighTime VAR word
rightHighTime VAR word
lowTime VAR word

leftHighTime = 1000
rightHighTime = 500
lowTime = 20

low 12
low 13
loop:
      pulsout 12,leftHighTime
      pulsout 13,rightHighTime
      pause lowTime
goto loop
```

Se inverti i valori di *rightHighTime* (non più 500, ma 1000) e *leftHighTime* (500, invece che 1000), vedrai il robot andare all'indietro. D'altra parte, attribuendo a entrambe le variabili di tempo high il valore intermedio 750, il robot resta fermo; con entrambe a 500, il robot ruota su se stesso in senso antiorario; mentre, con entrambe a 1000, gira su se stesso, ma in senso orario. Comunque sia, può darsi che uno dei due motori risulti più veloce dell'altro e che la traiettoria del robot, dunque, non sia precisa. In tal caso, bisogna procedere alla calibrazione. Consideriamo la variante del listato con entrambe le variabili a 750 (che dovrebbe arrestare il robot): se il robot si muove, bisogna variare il valore di uno dei due tempi high (procedendo con piccole variazioni, ad esempio di 1 in 1): ad esempio, se con *rightHighTime* = 750 la ruota del motore destro (collegato a P13) gira in senso orario, dovrai incrementare il valore a 751, 752, 753 ... fino a che si arresta (diciamo a 754). Quindi dovrai verificare il valore di pulsout necessario per rimettere in moto la ruota nel senso inverso: dovrai cioè continuare a incrementare pulsout (754, 755, 756 ...) finché la ruota non parte (diciamo a 756). Solo allora, infine, potrai calcolare la media dei due valori trovati: (754+756)/2=755, cioè il valore effettivo del pulsout di arresto della ruota. Se invece la ruota avesse girato in senso antiorario, avresti dovuto procedere diminuendo via via il valore di pulsout. Lo stesso vale poi per la seconda ruota.