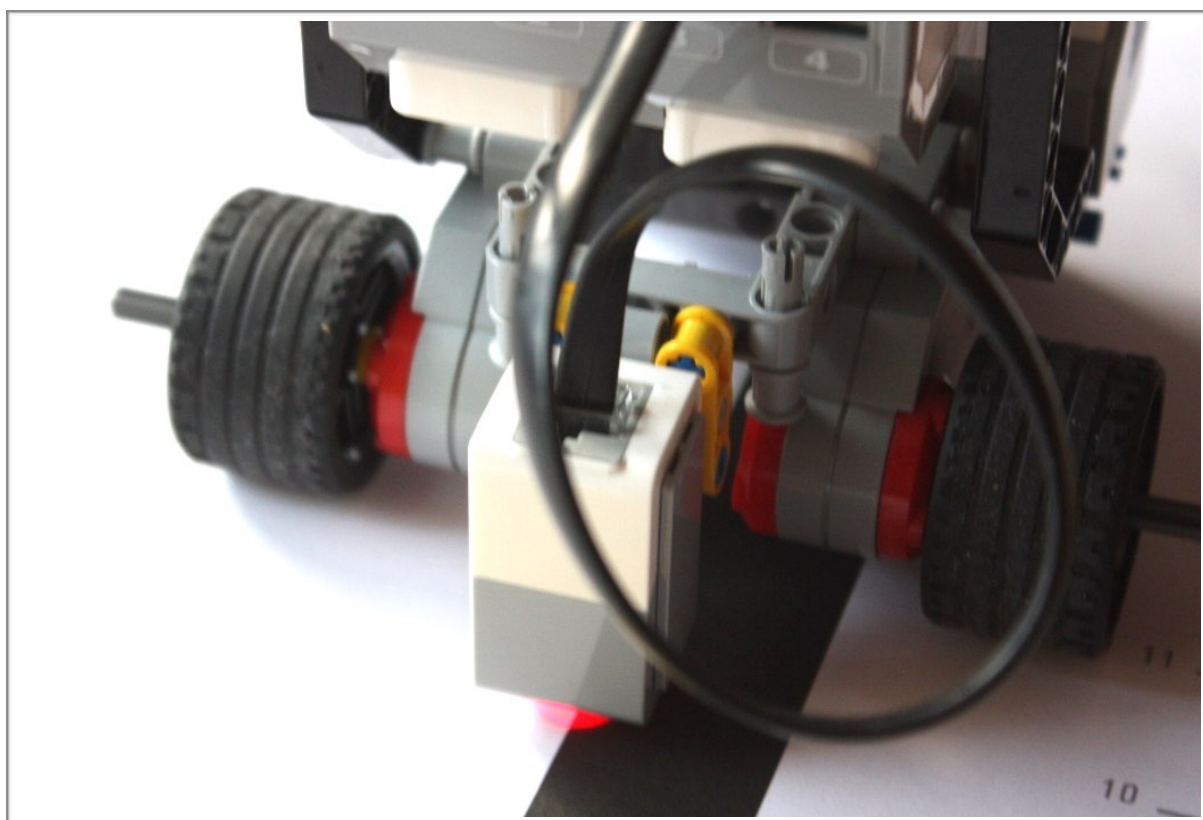


Line follower - Open Roberta

Il robot insegue una linea!



Lorenzo Daidone

Agosto 2016

Introduzione

Un robot line follower è il progetto ideale da cui partire per affrontare diverse problematiche sia hardware che software. Per prima cosa è opportuno definire i due termini. Per hardware si intendono tutte le componenti “tangibili” di un robot. Con il termine software si indicano l’insieme di programmi e procedure utilizzati per far eseguire ad un robot un determinato compito. Un robot *line follower*, *line*=linea e *follower* da *to follow*=seguire; è un dispositivo atto all’inseguimento di una linea.



Risorse hardware e software

- Computer connesso ad una rete wireless
- Kit Lego Mindstorm EV3 necessariamente completo di:
 - ✦ Brick EV3 (con batteria/e)
 - ✦ Due *big motor* Lego©
 - ✦ Sensore di colore (=colour sensor)
 - ✦ Sensore di tatto (=touch sensor)
 - ✦ Cavi per il collegamento dei motori e dei sensori
 - ✦ Vari pezzi lego (ruote, connettori, assi, ecc...)
- Open Roberta Starter Kit (dongle wi-fi, micro SD con firmware)
- Nastro isolante, preferibilmente di colore nero opaco


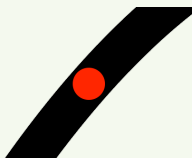
Funzionamento

Esistono diversi modi per costruire un *line follower*.

La soluzione che noi proponiamo è con solo un sensore, il robot quindi, avanzerà a cavallo della linea.

Nella fase di progettazione bisogna decidere se il *line follower* seguirà il confine destro o sinistro, rispetto al senso di marcia.

*Esempio con un line follower destro. Superficie chiara e linea nera.
Il punto rosso indica l'area di rilevamento del colore.*

Caso	Colore rilevato	Comportamento
	Bianco	Il robot ruota a destra finché rileva il colore nero.
	Nero	Il robot ruota a sinistra finché rileva superficie chiara.

Componenti elettroniche del robot

Il “cervello” del robot è il *brick*. Ad esso possono essere collegati al massimo quattro sensori e quattro motori. È dotato di schermo monocromatico 178x128 pixel, 6 pulsanti, porta USB A, bluetooth. Supporta anche una connessione wi-fi con apposito dongle, fornito nell'Open Roberta Starter Kit.

Le porte destinate al collegamento dei motori si trovano nella parte superiore del dispositivo e sono identificate con lettere dalla A alla D. Le porte destinate al collegamento dei sensori si trovano nella parte inferiore e sono numerati progressivamente dal numero 1 al 4. I cavi, avendo sempre lo stesso connettore, sono indifferenti sia per motori che per trasduttori. Uno degli errori più comuni, che possono far perdere molto tempo durante la fase di collaudo, è quello di non

aver collegato il cavo fino al fondo della porta; il caratteristico “click” darà conferma della corretto inserimento.

Il movimento dell'intero robot è affidato ai *big motor LEGO® EV3*. Ne servono solamente due perché le due ruote devono essere indipendenti, mentre il terzo punto d'appoggio è la *caster ball*, montata al centro, nella parte posteriore della struttura. La *caster ball* riduce l'attrito della la superficie su cui è contatto, agevolando il movimento dell'intera struttura.







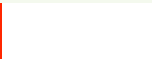



Ciò che abbiamo fin ora chiamato “motore” in realtà è un sistema più complesso. Il *big motor* è composto da: ingranaggi per aumentare la coppia motrice, *encoder*, un circuito di controllo e chiaramente, un motore.

L'encoder è la parte del sistema in grado di misurare la posizione dell'asse e garantire all'utente, mediante apposito controllo automatico, la rotazione desiderata. Ad esempio, se non fosse per questa parte fondamentale si potrebbe solamente regolare la rotazione del motore con un parametro **tempo** e non in **gradi/rotazioni**.

Per rilevare la linea sulla superficie si utilizza il sensore di colore. Il sensore è composto a sua volta da due apparati: emettitore e ricevitore. L'emettitore proietta i tre colori primari: rosso, verde e blu (RGB), il ricevitore ne misura l'intensità della luce riflessa. A seconda dell'intensità rilevata per ogni componente emessa, si potrà determinare con buona precisione il colore dell'oggetto sottostante al ricevitore.

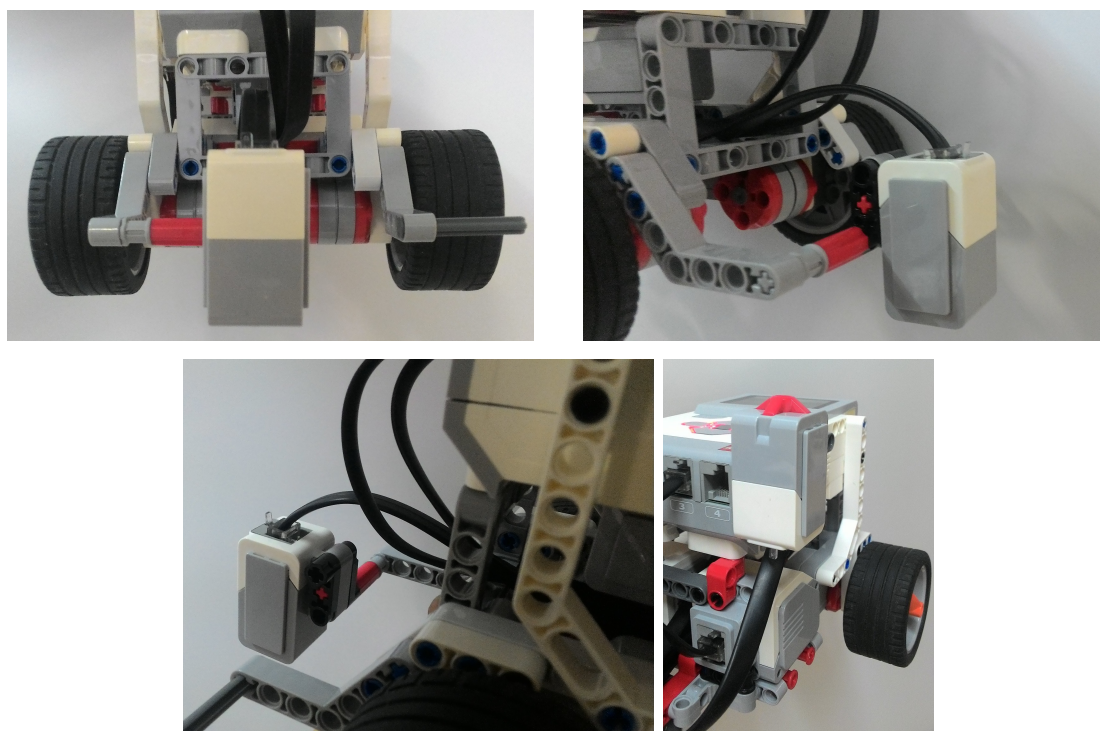
Nell'ambiente di programmazione Open Roberta Lab, la lettura del sensore viene processata per assimilare il colore riflesso dell'oggetto a solo 7 colori discreti, o eventualmente a nessun colore.

Nessun colore	Nero	Blu	Verde	Giallo	Rosso	Bianco	Marrone
							

Il sensore di tatto è opzionale, ma è stato aggiunto per rendere pratico il collaudo del software. Al momento, in Open Roberta Lab non è previsto un pulsante con cui interrompere l'esecuzione del programma. La funzione del sensore di tatto è, alla sua pressione, far giungere il programma al termine.

Costruzione della struttura

Per la costruzione della struttura si può seguire il manuale d'istruzioni fornito nel kit (da pag. 10 a 38 e da 69 a 71), altrimenti online si possono trovare molte altre soluzioni costruttive. Il telaio preso in considerazione nel tutorial ha il *color sensor* posto centralmente e il *touch sensor* sul lato.



Il sensore, per rilevare con precisione il colore della superficie deve essere montato a pochi millimetri di distanza dal suolo.

La tabella mostra i collegamenti.

Porta	Trasduttore
A	/
B	Motore sinistro
C	Motore destro

Porta	Trasduttore
D	/
1	Touch sensor
2	/
3	Color sensor
4	/

L'ordine di collegamento dei motori e sensori non è tassativo, si può modificare nella scheda "Robot configuration EV3basis" dell'Open Roberta Lab. Se si lascia l'ordine indicato in tabella non si dovrà apportare alcuna modifica alla scheda.

Operazioni preliminari

Open Roberta Lab è un ambiente di programmazione sviluppato dal Fraunhofer IAIS. Ogni icona ha un compito specifico e permettono di svolgere calcoli, confronti, gestire sensori, compiere azioni, scrivere un testo ecc... L'interfaccia utente comunica col robot utilizzando una connessione wireless.

In condizioni normali, all'accensione del brick EV3, viene caricato il sistema operativo integrato dalla casa madre. Open Roberta invece, possiede un suo sistema operativo, contenuto nella micro SD, che oltre a rendere possibile la comunicazione con l'ambiente di programmazione, possiede molte peculiarità.

Inserire la micro SD e il dongle wi-fi negli appositi slot. Accendere il brick premendo per circa un secondo il tasto centrale. La pulsantiera si illuminerà di rosso.

Alla prima accensione l'installazione del sistema operativo potrebbe richiedere fino a otto minuti. Una volta terminato il processo di installazione, la schermata finale somiglierà a questa:



Ora occorre connettere il brick alla rete wi-fi.

Spostarsi sull'icona "Wifi" del menu e premere il tasto centrale.

Verrà fatta una scansione delle reti wi-fi. Muoversi con i pulsanti per selezionare il proprio *access point* e confermare premendo il pulsante centrale. A questo punto, se necessario, inserire la password.

Nella schermata d'inserimento password sono presenti lettere, numeri e caratteri speciali. Nella parte bassa della schermata si possono trovare le lettere U, l, x, D. Ecco la funzione di ciascuna:

- **U**: lettere maiuscole
- **l**: lettere minuscole
- **x**: elimina l'ultimo carattere digitato
- **D**: conferma password

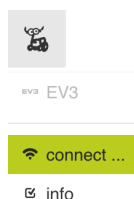
Al termine dell'inserimento dare conferma con la lettera **D**; il processo potrebbe richiedere alcuni istanti.

Se l'operazione è andata a buon fine, nell'angolo in alto a destra dovrebbe comparire l'icona *wifi* e il rispettivo indirizzo IP (nell'esempio 10.116.20.173).



A questo punto resta solo il collegamento di Open Roberta Lab con il brick.

1. Aprire un browser dal proprio computer, collegarsi al sito <https://www.open-roberta.org/willkommen/> e fare click su "Starten". Si aprirà in una nuova scheda l'ambiente di programmazione.
2. Ritornare al brick, muoversi nel menu con le frecce fino all'icona "Open Roberta Lab" e dare conferma con il tasto centrale. Verrà mostrata una password.
2. Nella schermata del browser, fare click sul caratteristico logo di O.R. e poi su "connect ..."



3. Inserire la password nel campo di testo. Al termine premere su OK.

L'emissione di un suono darà conferma del collegamento fra i due dispositivi.

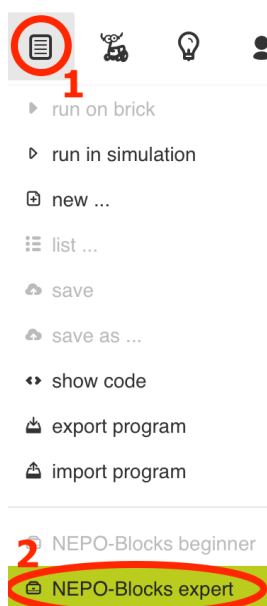
Il programma

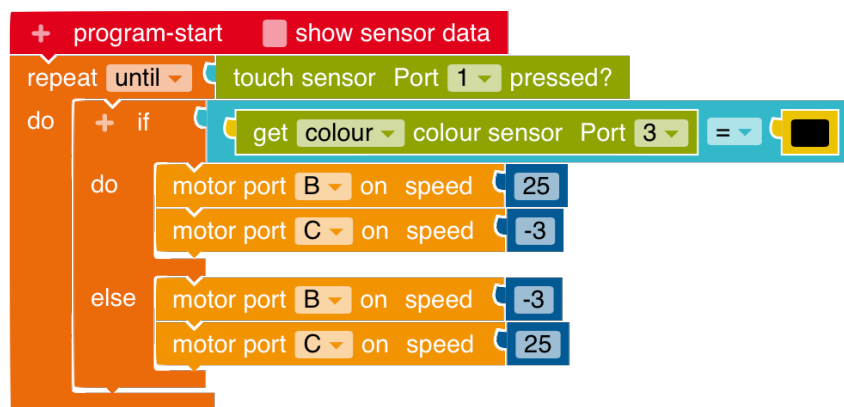
Generalmente, il programma di un line follower con solo un sensore di luce/colore funziona nel seguente modo: viene effettuata una lettura del colore sottostante al sensore. Se il colore è quello della linea (ad es. nero) il robot curva da una parte, se il colore è un altro il robot curva dalla parte opposta.

La curva non deve essere netta, ma anzi lieve, per consentire nel frattempo al robot di avanzare nel percorso.

In aggiunta, come anticipato in precedenza, prima di effettuare il controllo del colore viene letto lo stato del sensore di tatto. Nel caso di una pressione da parte dell'utente, non viene più eseguito il ciclo per il movimento e l'esecuzione salta alla successiva icona. Siccome non sono presenti altre icone, il programma termina.

Di default alcune icone utilizzate nel programma del line follower non sono disponibili, per renderle visibili andare sotto la voce "edit" (prima icona in alto a sinistra) e fare click su "NEPO-Blocks expert".





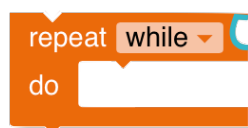
I tipi di blocchi sono differenziati per colore.

- **Arancione chiaro**: icone relative al movimento
- **Arancione scuro**: gestione del flusso del programma
- **Azzurro**: operazioni logiche
- **Verde**: sensori
- **Giallo**: colori

La prima icona del programma è di colore rosso. Essa permette di fare configurazioni iniziali, che in questo caso non sono necessarie.

Come anticipato in precedenza, un ciclo (repeat until ..., do) regolerà il flusso del programma, affinché venga interrotto alla pressione del bumper.

Il blocco *repeat ... do* funziona nel modo seguente:



- Tipo **while**: finché la condizione è vera vengono eseguite ripetutamente le icone contenute all'interno del ciclo
- Tipo **until**: finché la condizione è falsa vengono eseguite ripetutamente le icone contenute all'interno del ciclo

La condizione può assumere solamente due stati: **vero** (1) o **falso** (0). Assume stato logico 1 (vero) quando un confronto è verificato (ad es. $5 > 2$? SÌ, quindi **vero**); assume stato logico 0 quando non è verificato.

In questo caso è usato un ciclo di tipo **until**, perché di norma, quando non è premuto il sensore di tatto, la lettura relativa produce una condizione **falsa**; nel caso il pulsante venga premuto, produce una condizione **vera**.



La direzione della curva viene stabilita dal blocco *if - else*. Il valore della condizione in questo caso è data dal confronto fra il colore letto dal sensore e il colore reale della linea.

Se (*=if*) il colore rilevato è uguale al colore della linea, il robot curva a destra altrimenti (*=else*) curverà a sinistra.

Per stabilire le velocità dei motori si utilizza il blocco sotto rappresentato, individuabile sotto *Action > Move*.



Facendo click sul menu di selezione, si stabilisce a quale motore applicare la velocità desiderata. La velocità è indicata nel blocco di colore blu ed è espressa in percentuale da -100 a 100; i numeri negativi equivalgono ad una rotazione nel senso opposto.

Percorso

Nel disegno e nella costruzione del percorso su cui il robot svolgerà le sue funzioni è completamente a discrezione dell'utente. Il più semplice sia da disegnare che da far seguire al proprio robot è un circuito chiuso con curve morbide. Curve nette, invece, mettono generalmente a dura prova la robustezza del programma. In alcune competizioni il percorso può prevedere incroci, interruzioni della linea ed ostacoli su essa.

Il tutorial prende in considerazione un circuito chiuso con curve morbide.

Due consigli:

1. La linea potrà essere realizzata con del nastro isolante o con vernice; ciò che è importante è scegliere superfici opache in quanto luci secondarie riflesse potrebbero costituire un “rumore” e falsare il rilevamento del sensore.
2. Scegliere colori contrastanti per la superficie e la linea, ad esempio superficie bianca e linea nera.

Collaudo e taratura

Il collaudo del software in questo caso può essere effettuato confrontando il proprio programma con quello alla sezione “Software”.

La taratura invece deve essere svolta necessariamente in modo empirico, poiché essa dipende da molti fattori, ad esempio: forza attrito fra le superfici, peso della struttura, illuminazione della stanza, ecc...

Inizialmente si pongono le velocità dei due motori una la metà dell'altra, avendo l'accortezza di scegliere velocità relativamente basse. È sempre meglio dare tempo al processore del robot di verificare più volte, in uno spazio ristretto, se esso si trova sulla linea o sulla superficie. (esempio 30 e 15)

Ricordare sempre di modificare in modo opposto le velocità sia della curva a destra e della curva a sinistra.

Testare i valori premendo il pulsante **run**.

Probabilmente il risultato ottenuto non andrà bene. Ridurre le velocità massime e minime, eventualmente inserendo anche valori negativi (la ruota girerà nel senso opposto).

Il processo di taratura probabilmente andrà ripetuto molte volte.

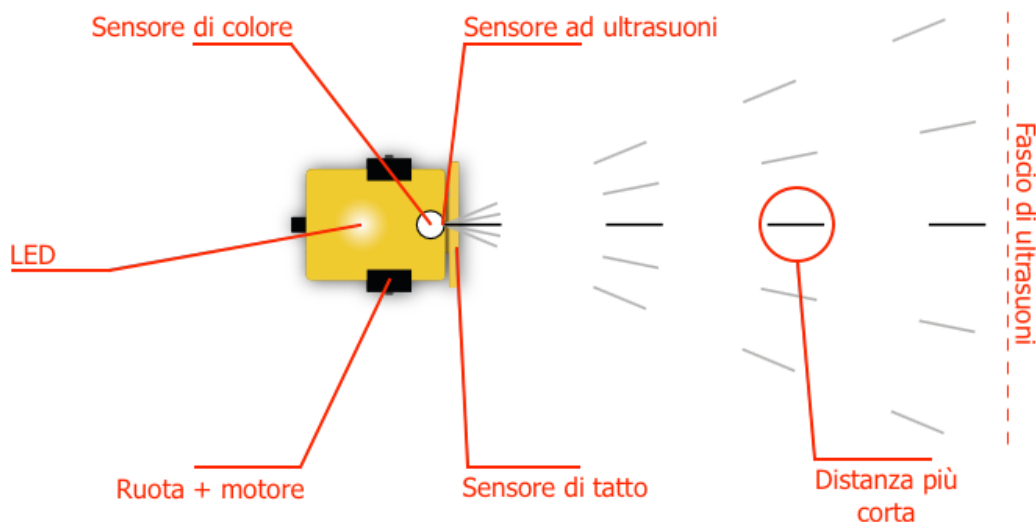
Ciò che darà conferma di aver realizzato un buon line follower sarà il repentino cambiamento di direzione. Il frequente cambiamento della direzione è indice di un controllo effettuato molte volte, garantendo all'utente una buona precisione.

Simulazione

Open Roberta Lab offre anche la possibilità di simulare il programma su diversi scenari e con un robot in 2D.

Non tutti i blocchi però possono essere simulati, a causa della semplice configurazione del robot disponibile nell'ambiente di simulazione.

Il robot 2D è così composto:



Sensor 1	Touch sensor
Sensor 2	Gyroscope
Sensor 3	Color sensor
Sensor 4	Ultrasonic sensor
Motor A	/
Motor B	Big motor (right)
Motor C	Big motor (left)
Motor D	/

Si può modificare l'ordine e il tipo di sensori/motori utilizzati nella scheda "Robot configurazione EV3basis" introdotta precedentemente.



Per avviare la simulazione premere il pulsante in basso a destra **"run in simulation"**.



Una volta premuto il pulsante si aprirà il pannello di simulazione. Nell'angolo in alto a destra sono mostrati una serie di dati sulla simulazione e sul robot simulato.

- **FPS:** la quantità di frame simulati in un secondo
- **Time:** tempo trascorso dall'avvio della simulazione
- **Robot X** e **Robot Y:** coordinate del robot
- **Robot θ :** angolo, espresso in gradi che indica la rotazione compiuta rispetto l'asse X
- **Motor left:** potenza del motore sinistro
- **Motor right:** potenza del motore destro
- **Touch sensor:** stato del sensore di tatto (posto anteriormente)
- **Light sensor:** stato del sensore di luce
- **Ultra sensor:** distanza rilevata dal sensore ad ultrasuoni, espressa in centimetri
- **Color sensor:** colore rilevato dal sensore (posto anteriormente, rivolto verso la superficie)

Per ritornare alla schermata in cui è visualizzato solo l'ambiente di programmazione, fare click sulla freccia nell'angolo in basso a destra.



Problematiche simulazione e realtà

Se si confronta il comportamento del robot reale con quello simulato si nota un comportamento decisamente differente.

Considerando, ad esempio, il programma del *line follower*, con tarature per robot reale, il movimento nella simulazione risulta essere molto più lento. Questo perché la velocità di calcolo della simulazione è decisamente maggiore rispetto a quella del microprocessore del brick. In uno stesso spazio vengono fatte molte più verifiche al secondo.

Per ottenere in simulazione un comportamento simile a quello della realtà, bisogna introdurre un blocco di ritardo alla fine del ciclo.



Il blocco wait si trova nelle icone *Control* > *Wait* > *wait ms*.

L'icona wait introduce un ritardo fra l'esecuzione del blocco che la precede e il blocco che la segue. Il ritardo è espresso in millisecondi.