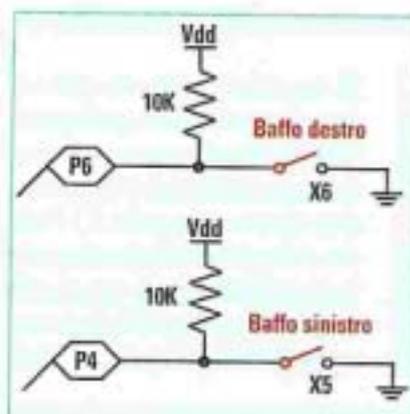




I sensori di contatto

I pezzi allegati servono per il montaggio dei sensori di contatto. Data la loro forma e la funzione che svolgono, tali sensori sono detti "baffi": similmente alle vibrisse (baffi) di un gatto, infatti, questi sensori permettono al robot di avvertire la presenza di eventuali ostacoli sul suo percorso. I baffi sono costituiti da due estremità di plastica collegate a una lamina flessibile d'acciaio. Questa lamina, montata su un distanziale cilindrico di plastica, è fissata per mezzo di una vite alla scheda madre del robot così da essere inserita tra i due contatti di ciascun connettore (che sono indicati sulla scheda madre dalle diciture X5 e X6) e in modo da non toccare quello posteriore. Quando uno dei due baffi incontra un ostacolo, la lamina, grazie alla sua flessibilità, si piega, toccando così il contatto posteriore di ciascun connettore. Ciascuno dei due contatti posteriori è collegato a un circuito della scheda madre: il movimento dei baffi e la conseguente aderenza tra il contatto e la lamina d'acciaio determinano un cambiamento in tale circuito, realizzando in pratica quello che viene schematizzato in figura. I due connettori per i sensori di contatto sono collegati alla tensione Vdd



● **Sopra.** Gli schemi circuitali dei due sensori di contatto (i "baffi") del robot.

e alle due porte P4 e P6 del microcontrollore. I baffi hanno la funzione degli interruttori: quando la lamina è in posizione "di riposo" il ramo circuitale destro risulta

aperto e la tensione alla porta del microcontrollore è di 5 V, un valore logico alto. Quando invece la lamina d'acciaio si appoggia ad almeno uno dei contatti posteriori dei connettori, il ramo circuitale destro viene chiuso, portando così l'anello metallico presente sulla scheda madre (che è in contatto con la lamina tramite la rondella e la vite di fissaggio) alla tensione di massa: si realizza così un corto circuito che porta a zero, cioè al valore logico basso, la tensione alla porta del microcontrollore. Presto vedremo come leggere il valore logico delle due porte. Potrai così stabilire quale baffo ha urtato l'ostacolo e programmare il movimento per poterlo superare.

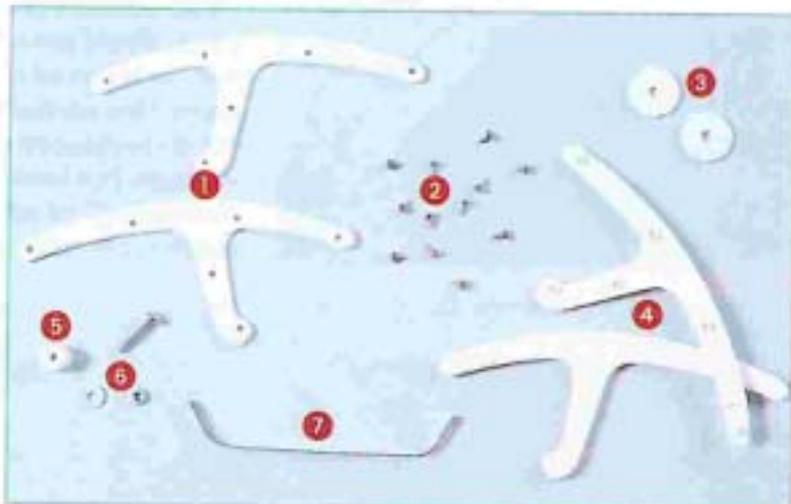
Le fasi di montaggio

Le operazioni di montaggio che stai per eseguire doteranno il robot di una nuova tipologia di accessorio,

cioè i sensori di contatto. I baffi allegati possono essere di colore bianco (come nella foto) oppure grigio metallizzato.

L'ELENCO DEI PEZZI

- 1 n. 2 parti inferiori dei baffi in plastica
- 2 n. 12 viti da 4,5 mm
- 3 n. 2 rondelle in plastica da 1,9 cm
- 4 n. 2 parti superiori dei baffi in plastica
- 5 distanziale cilindrico in plastica
- 6 vite, rondella e dado M3 per il fissaggio dei baffi sulla scheda madre
- 7 lamina preformata in acciaio inossidabile



A Disponi sul piano di lavoro le due parti superiori dei baffi in modo che appoggino sul lato piatto (i cilindri a sbalzo devono essere visibili). Fai attenzione che le estremità di ogni baffo con l'invito per la rondella (cerchiato in rosso) siano rivolte verso l'esterno.



B Inserisci le estremità della lamina in acciaio nelle guide predisposte sui baffi; esercita una pressione con le dita per assicurare l'incastro della lamina nelle guide. **C** Posiziona una delle rondelle in plastica sul cilindro (indicato dalla freccia rossa) posto all'estremità di un baffo; fai lo stesso con la seconda rondella sull'altro baffo.



D Sovrapponi la parte inferiore di uno dei baffi a quella superiore corrispondente, in modo da far coincidere i sei cilindri di questa con le sedi per il loro incastro dell'altra. **E** Esercita una delicata pressione con le dita per assicurare l'incastro delle due parti.



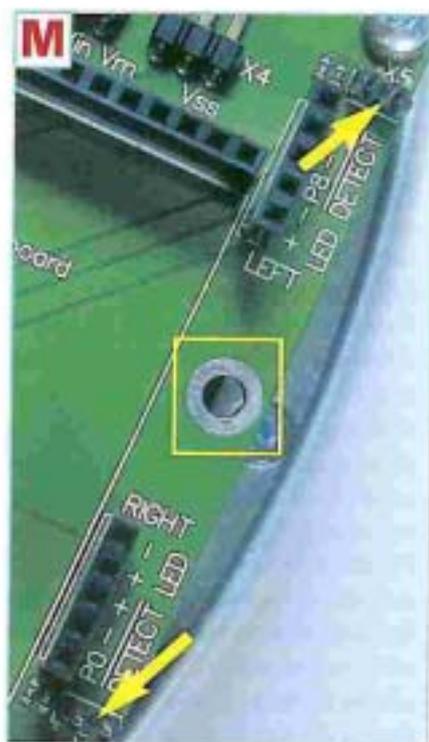
F Dopo aver montato allo stesso modo le due parti dell'altro baffo, disponi le sei viti nelle apposite sedi (indicate dalle frecce rosse) sulla parte inferiore di uno dei due baffi e avvitalo con il cacciavite a stella. Ripeti l'operazione sul secondo baffo con le altre sei viti. **G** Ecco come appare il lato inferiore dei baffi una volta portato correttamente a termine il loro montaggio. **H** Il lato che sarà visibile quando monterai i baffi sul robot è invece quello superiore, che appare completamente liscio.



I Mantenendo a vista il lato superiore dei baffi, inserisci la lamina nel taglio del distanziale cilindrico (come evidenziato nel particolare del tondo).



L Inserisci la vite con la rondella nell'apposito foro del distanziale cilindrico. Come avrai notato, la lamina sporge leggermente dalla guida del distanziale, creando un contatto con la rondella e, dunque, con la vite: solo in questo modo la lamina potrà realizzare il circuito con la scheda madre. **M** Individua sulla scheda madre il foro di passaggio della vite (riquadrate in giallo) e le coppie di contatti dei connettori X5 e X6 (indicate dalle frecce gialle) tra i quali si inserirà la lamina dei baffi.



N A questo punto devi fissare i baffi alla scheda madre. L'operazione, ovviamente, deve essere effettuata in modo tale che i baffi sporgano verso l'esterno del robot e non siano invece rivolti verso il centro del telaio. Infila la vite del blocco distanziale-baffi nel foro che hai appena individuato sulla scheda madre stessa.



O Controlla che la lamina dei baffi sia posizionata tra i contatti dei connettori X5 e X6 (riquadrate in giallo). Nota che la lamina si appoggia, toccandoli, solo ai contatti anteriori dei connettori. I contatti posteriori e la lamina, infatti, dovranno toccarsi soltanto quando il baffo urterà un ostacolo. **P** Centra i baffi rispetto alla scheda madre in modo che risultino equidistanti dal distanziale, poi posiziona il dado sotto la scheda (come evidenziato nel tondo) e fissa i baffi avvitando la vite.



Le fasi di programmazione

Quello di variabile è sicuramente uno dei concetti fondamentali per la programmazione. All'interno di un programma le variabili hanno lo stesso ruolo che rivestono in un'espressione algebrica: sono cioè nomi simbolici che rappresentano determinati valori, solitamente numerici. In particolare, però, nella programmazione, una variabile fa riferimento a una

determinata zona della memoria centrale (nel nostro caso la RAM del microcontrollore), contenente una sequenza di bit che viene letta e che può essere modificata durante l'esecuzione del programma.

Per capire il funzionamento delle variabili, digita nell'area di editing il listato del programma riportato a destra. Quindi collega il robot al PC e accendilo, poi premi il tasto **Run** perché il programma venga eseguito. Questo programma apre una finestra di debug (in basso a destra) che visualizza le variabili e le operazioni effettuate con esse. Osservando il listato, noterai che, dopo la direttiva iniziale PBASIC, il programma è stato diviso in tre parti, ciascuna contrassegnata da un commento tra alcuni trattini ('---commento---'); questa struttura non è obbligatoria, ma caratterizza un programma ben scritto, è dunque buona norma utilizzarla. Anche la **spaziatura multipla** tra le istruzioni è una norma di buona scrittura: per programmare correttamente, infatti, sarebbe sufficiente un solo spazio tra le istruzioni; inserirne di più, però, permette di 'indentare' il listato (disporlo con ordine nell'area di editing). Torniamo al listato: **la prima parte riguarda la dichiarazione delle variabili** che saranno usate. La sintassi della dichiarazione esige il nome con cui si intende identificare e richiamare la variabile nel programma (**variabile_1**, **x**, **y** e **risultato**), seguito da **var** (che, in PBASIC, caratterizza la dichiarazione di variabili) e dalla dimensione (**bit**, **nib** ecc.), in termini di memoria occupata, che si intende adottare. È necessario che il nome scelto sia costituito da una serie di caratteri alfabetici e/o numerici (e non simbolici, come \$, %, +, *, &, riservati ad altre funzioni) e il primo deve essere alfabetico. Buona norma è che il nome abbia un legame di senso con il ruolo svolto dalla variabile.

Dimensione	bit	Numeri rappresentabili
bit	1	Da 0 a 1 ($=2^1-1$)
nib	4	Da 0 a 15 ($=2^4-1$)
byte	8	Da 0 a 255 ($=2^8-1$)
word	16	Da 0 a 65 535 ($=2^{16}-1$) o da -32 768 a 32 767

La dimensione riguarda il numero di bit e, dunque, lo spazio di memoria che deve essere riservato al contenuto di una variabile: maggiore è il numero che voglio rappresentare, maggiore deve essere il numero di bit a disposizione. Le possibili dimensioni (nella tabella qui sopra) sono bit, nib, byte e word. La dimensione **word**, oltre a essere la maggiore (fino a 65 535), permette anche di rappresentare numeri negativi (nel qual caso, però, si dimezza il valore massimo memorizzabile, riducendolo a 32 767): si potrebbe allora usare sempre questa dimensione ma, per non sprecare memoria utile, conviene scegliere sempre la minima dimensione possibile. Procedendo nel listato, **la seconda parte riguarda l'inizializzazione delle variabili**, cioè l'assegnazione dei rispettivi valori iniziali, tramite l'operatore = preceduto dal nome della variabile e seguito dal valore da assegnare.

```

BASIC Stamp - D:\Documents and Settings\Simone\My Documents\ROBOTICS\prog
File Edit Device Run Help
-----
variabili.b2 |
'($STAMP B52)
'----- dichiarazione -----
variabile_1   var   nib
x             var   byte
y             var   word
risultato     var   word
'----- inizializzazione -----
x = 127
y = 1329
variabile_1 = 5 * x - 631
'----- programma principale -----
debug DEC x, CR, CR
debug DEC ? x, DEC ? y, DEC ? variabile_1, DEC ? risultato, CR
risultato = (y / 3) * x
debug DEC ? risultato
1 1
  
```

Quest'ultimo può essere un numero o un'espressione numerica che contenga, a sua volta, altre variabili (è il caso di **variabile_1**, che contiene la **x**). Nel caso in cui il valore iniziale di una variabile non venga specificato, tale valore viene automaticamente inizializzato a 0. **La terza sezione del listato, infine, contiene il programma principale.** L'istruzione **debug DEC x, CR, CR** visualizza nell'apposita finestra (**debug**) il valore decimale (**DEC**) della variabile **x**, andando poi a capo (**CR**) due volte (prima riga a destra). Ogni volta che nel programma (sopra) ricorre l'istruzione **debug**, la finestra corrispondente (a destra) visualizza ciò che segue tale istruzione, ossia una stringa di testo o il contenuto numerico di variabili, di cui però va specificato il formato (in questo caso decimale). Nella seconda istruzione del programma principale, premettendo il carattere **?** ai nomi delle variabili (**x**, **y**, **variabile_1** e **risultato**), si visualizzano anch'essi, oltre al loro valore (nella finestra qui sopra, dalla riga in cui compare **x = 127** a quella con **risultato = 0**). Non essendo stata espressamente inizializzata, la variabile **risultato** vale 0. Nella successiva istruzione del listato, invece, alla variabile **risultato** viene assegnato, tramite l'operatore =, il valore $(y / 3) * x$. L'ultima istruzione visualizza nome (?) e valore, in forma decimale (**DEC**), di **risultato** (ultima riga qui sopra).

```

Debug Terminal #1
Com Port: Serial Port 1
Data File:
New Control:
TX: [ ] RX: [ ]
RTS: [ ] DSR: [ ]
127
x = 127
y = 1329
variabile_1 = 4
risultato = 0
risultato = 30201
  
```