

MODULO DI CONTROLLO



PL

MODULO DI CONTROLLO

Modulo di controllo
Modulo di Ingresso/Uscita
Modulo di Potenza
Modulo dei Sensori
Modulo Ottico
Telaio

Nella presente sezione viene spiegato in modo chiaro la metodologia da seguire per realizzare il corretto assemblaggio del robot. Sarà anche illustrata la programmazione del robot, con numerosi esempi, e saranno proposte delle idee da sviluppare ed applicare su Monty, il nostro robot intelligente.

SALDATURA

Insieme a queste pagine viene fornita la prima scheda di cui è formato il robot, la scheda di controllo. In questa scheda è necessario saldare i distinti componenti elettrici di Monty, è fondamentale che l'operazione di saldatura sia realizzata nella maniera adeguata.

In seguito, sarà spiegata qual è la tecnica idonea di saldatura che deve essere impiegata ogni volta per il montaggio dei componenti, nella scheda PCB (Printed Circuit Board o circuito stampato). I due elementi fondamentali che saranno utilizzati per saldare sono lo stagno ed il saldatore. Lo stagno è un metallo che fonde a temperatura non molto elevata. Il saldatore genera il calore necessario, e serve affinché lo stagno si fonda sopra la superficie di contatto che unisce il piedino di un componente con la scheda PCB. Dobbiamo cercare di evitare il contatto diretto fra il saldatore e lo stagno, perché lo stagno fonde molto velocemente, e si raffredda altrettanto velocemente quando si toglie il saldatore dalla superficie che si sta saldando. In questo modo il processo di solidificazione dello stagno non si realizza perfettamente, e con il passare del tempo potrebbero comparire problemi di conduttività. Questo tipo di saldatura viene denominata "fredda". Per sapere se abbiamo realizzato bene una saldatura, questa deve brillare alla luce, se così non fosse, è possibile che la saldatura sia fredda.



Il metallo con il quale vengono saldati i componenti elettronici sulla scheda del circuito stampato chiamato PCB (dall'inglese Printed Circuit Board, circuito stampato) è lo stagno. Per questo deve essere fuso utilizzando il saldatore.

I passi da seguire per ottenere una corretta saldatura sono:

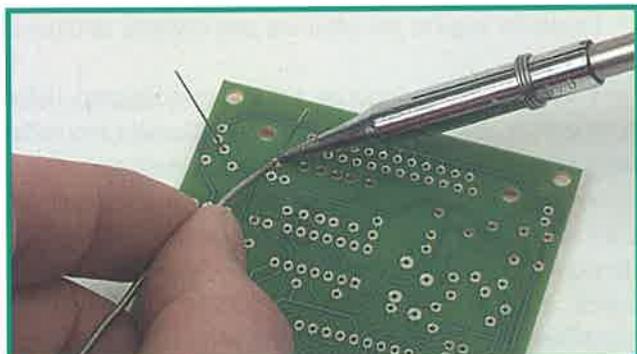
- Inserire il componente nel lato predisposto della scheda PCB, verificando che i terminali fuoriescano nella posizione dove si trova la piazzola per la saldatura.
- Avvicinare il saldatore alla piazzola da saldare, dove c'è il terminale del componente, riscaldare per un breve periodo e quindi avvicinare lo stagno. Il saldatore deve essere a punta fine come quello indicato nella foto. Riscaldare sino a quando lo stagno comincerà a fondere ed a formare un cono attorno alla superficie di contatto del componente con la scheda.
- In quell'istante ritirare il saldatore e lo stagno, lasciando che la saldatura si raffreddi in maniera naturale (non soffiare).
- Una volta che il componente è saldato e la saldatura si è raffreddata, dobbiamo tagliare con un tronchese la parte del terminale che fuoriesce, in modo che l'aspetto finale della saldatura sia a forma di cono.
- Più avanti sarà spiegato come si può togliere la saldatura, mostrando diversi attrezzi necessari per manipolare ed assemblare i componenti elettrici.

CONTENUTO DEL CD-ROM

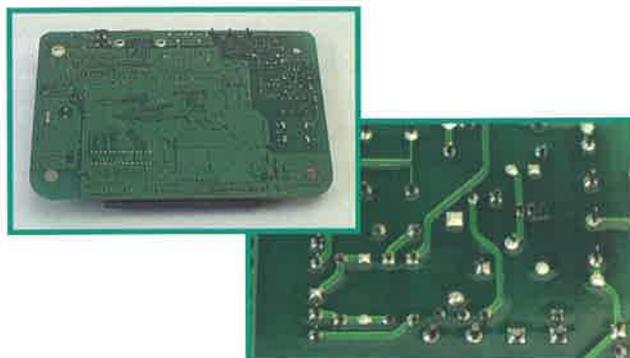
Nel CD-ROM allegato a questo fascicolo troverete una serie di video nei quali si possono osservare alcuni concorsi di microbotica. In queste manifestazioni, i micro-robots concorrono in diverse specialità, nelle quali essere il più "intelligente" è la chiave dell'esito. Nel CD-ROM viene incluso un software che sarà utilizzato per la programmazione del microcontrollore PIC, cervello del robot, compito che sarà spiegato più avanti e che potrà essere eseguito quando sarà stata assemblata la scheda di controllo.



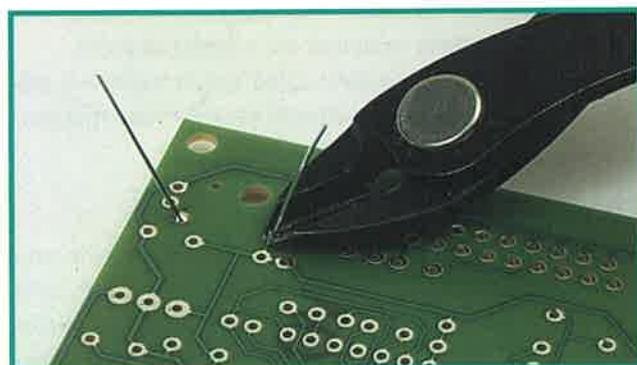
Il saldatore è un attrezzo che genera il calore necessario per poter fondere lo stagno. Ricordiamoci, quando eseguiremo le operazioni di saldatura, di evitare il contatto diretto fra lo stagno e la punta del saldatore.



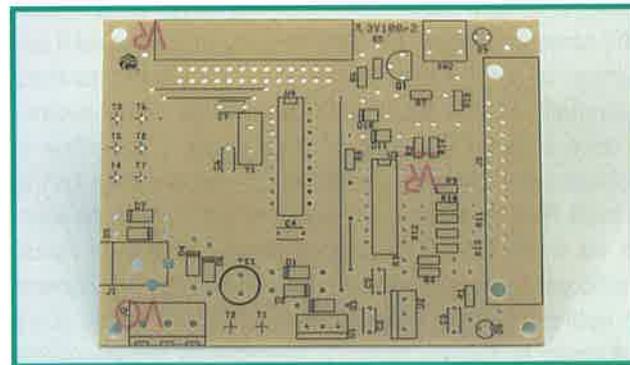
Per realizzare una saldatura idonea dobbiamo posizionare il saldatore in modo che tocchi il terminale del componente e la scheda contemporaneamente. Dopo dobbiamo avvicinare lo stagno dalla parte opposta del terminale, con questa procedura vengono raggiunti buoni risultati.



Per realizzare una buona saldatura bisogna:
 1. Inserire il componente nel lato componenti della scheda PCB.
 2. Riscaldare bene l'area con il saldatore.
 3. Fondere lo stagno fino a formare un cono.
 4. Spostare il saldatore ed aspettare che si raffreddi.
 Nella foto viene mostrata una scheda correttamente saldata.



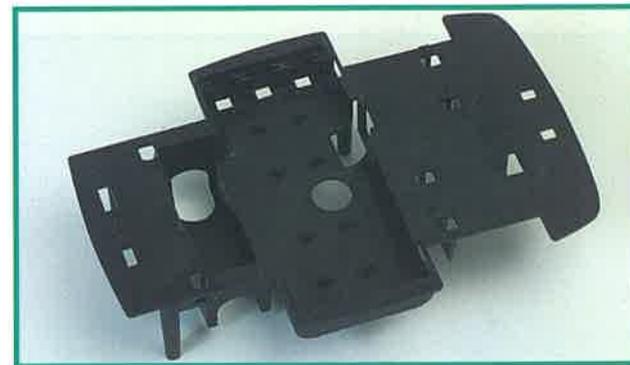
Una volta realizzata una saldatura dobbiamo aspettare che lo stagno si raffreddi naturalmente, evitando di soffiarci sopra. Quando la saldatura è fredda dobbiamo tagliare con un tronchesino il terminale che sporge dal componente.



La serigrafia di una scheda indica:
 1. Il nome assegnato al componente di quella scheda.
 2. Le dimensioni. 3. La sua polarità. 4. L'orientamento.
 Tutto questo serve per installare correttamente i componenti prima di essere saldati nella scheda.



La struttura meccanica più importante del robot è il telaio, nel quale saranno inseriti il motore, le ruote e le schede del circuito impresso. Avrà anche il compito di portare tutta la struttura del robot.



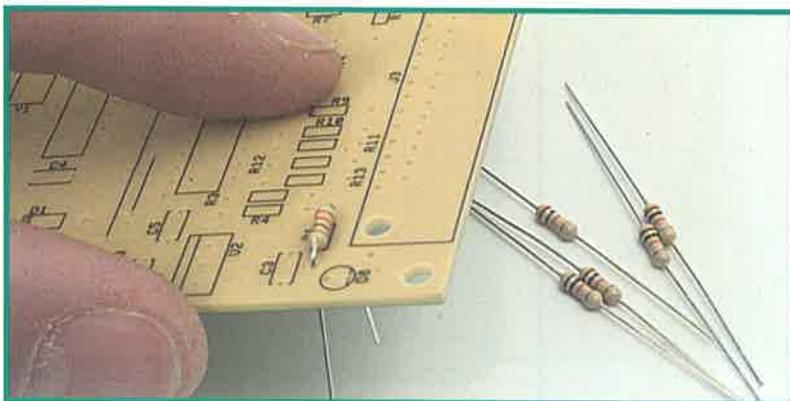
Il telaio di Monty è anche predisposto per l'inserimento di diversi sensori ottici e meccanici, come un altoparlante e le batterie necessarie per l'alimentazione del circuito.

1ª Cifra	2ª Cifra	Moltiplicatore	Tolleranza		
Nero = 0	Nero = 0	x1	<table border="1"> <tr> <td>Argento 10%</td> </tr> <tr> <td>Oro 5%</td> </tr> </table>	Argento 10%	Oro 5%
Argento 10%					
Oro 5%					
Marrone = 1	Marrone = 1	x10			
Rosso = 1	Rosso = 2	x100			
Arancione = 3	Arancione = 3	x10.00			
Giallo = 4	Giallo = 4	x10.000			
Verde = 5	Verde = 5	x100.000			
Blu = 6	Blu = 6	x1.000.000			
Viola = 7	Viola = 7	Oro: 10			
Grigio = 8	Grigio = 8	Argento:100			
Bianco = 9	Bianco = 9				

Inizieremo assemblando le resistenze. Il valore delle resistenze viene indicato sulle stesse con delle bande colorate. Per poter leggere i valori viene utilizzata la tabella dei codice colori, ma per leggerla correttamente è necessario mettere la resistenza orizzontale con la banda dorata a destra.

Esempio: per ottenere il valore della resistenza dell'immagine dobbiamo leggere le bande colorate.

1ª banda = marrone..... 1
 2ª banda = nero..... 0
 3ª banda = arancione..... x1.000
 Totale = 10x1.000..... 10 K Ω

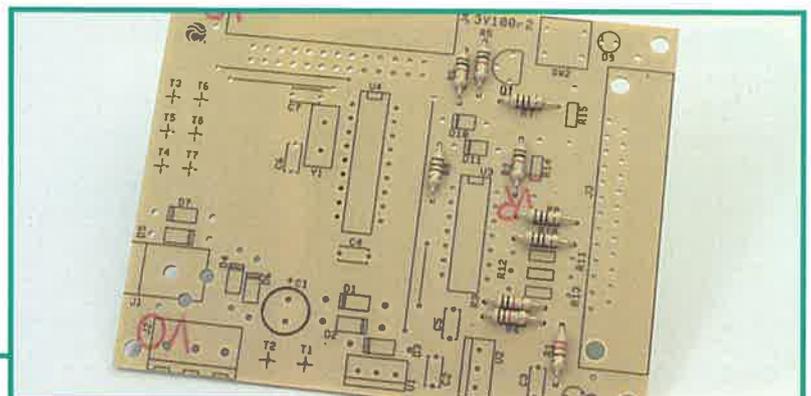


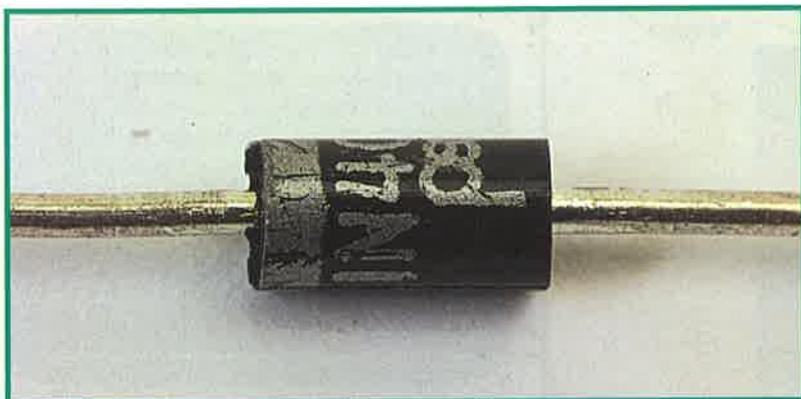
I terminali della resistenza vengono piegati ed inseriti nella scheda nel posto indicato dalla serigrafia, tutte nello stesso senso. Le resistenze devono rimanere a filo della scheda.

Le resistenze che dobbiamo saldare sono:

R1: 330 Ω = arancione - arancione - marrone.
 R2 - R3 - R4 - R5 - R6: 10 K Ω = marrone - nero - arancione.
 R7 - R8 - R9 - R10: 100 Ω = marrone - nero - marrone.

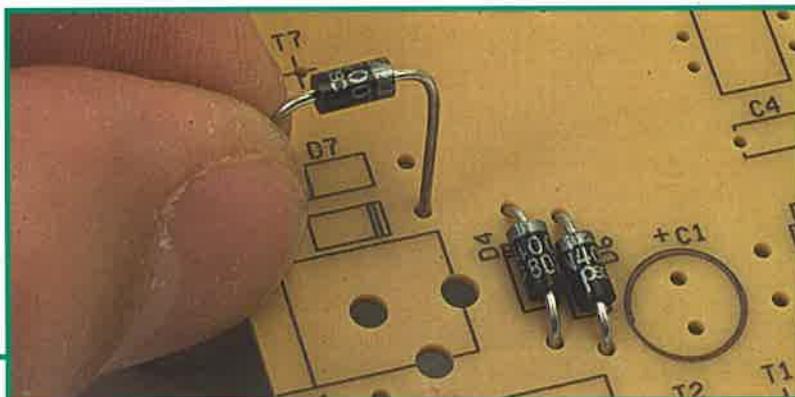
Il risultato, dopo la saldatura, deve essere come nella fotografia.



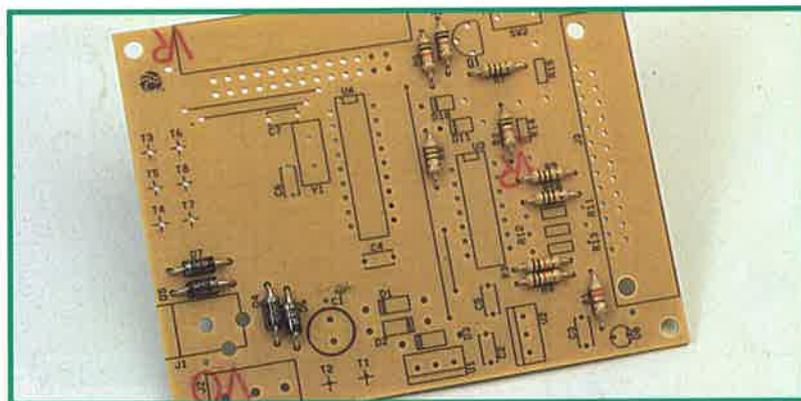


A differenza delle resistenze i diodi sono provvisti di polarità, e vanno collocati secondo un determinato verso.
Nella serigrafia della scheda i diodi sono indicati come D4, D5, D6 e D7.

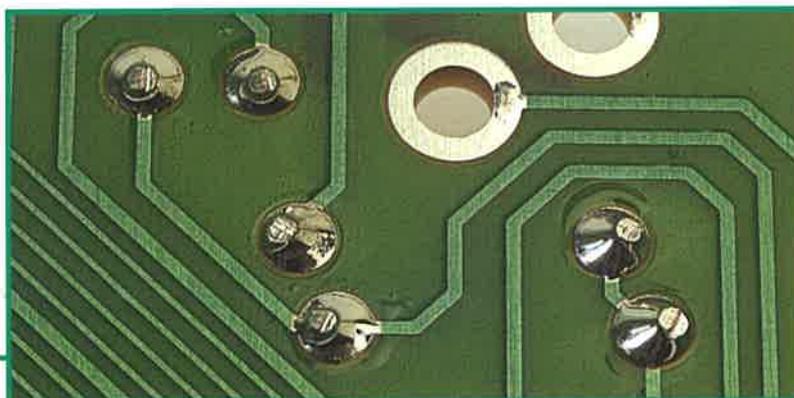
Faremo coincidere la banda grigia del diodo con la striscia nera della serigrafia sulla scheda. Una volta collocato, salderemo il diodo.



Terminate tutte le saldature i componenti devono apparire come si vede nell'immagine: a filo della scheda e per il verso corretto.



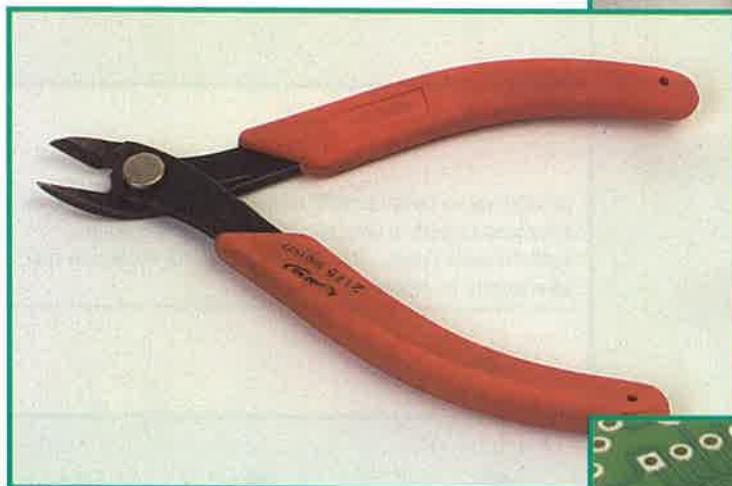
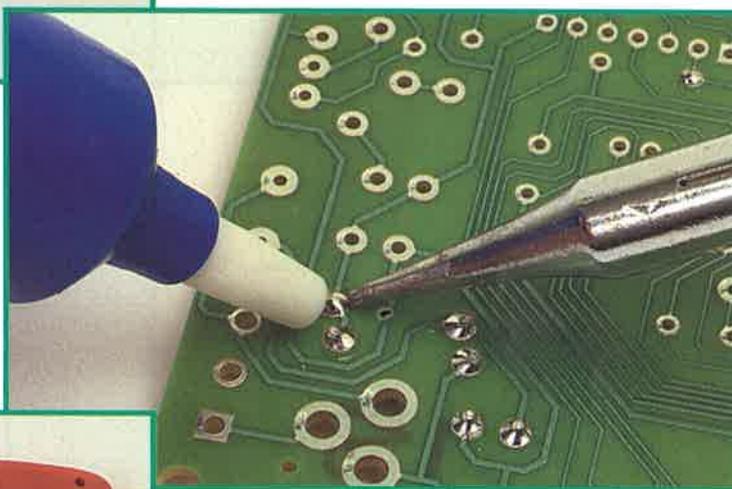
Una saldatura di buona qualità deve avere la forma di un cono ed essere brillante. Per questo si eviterà, per quanto possibile, di toccare lo stagno direttamente con la punta del saldatore.





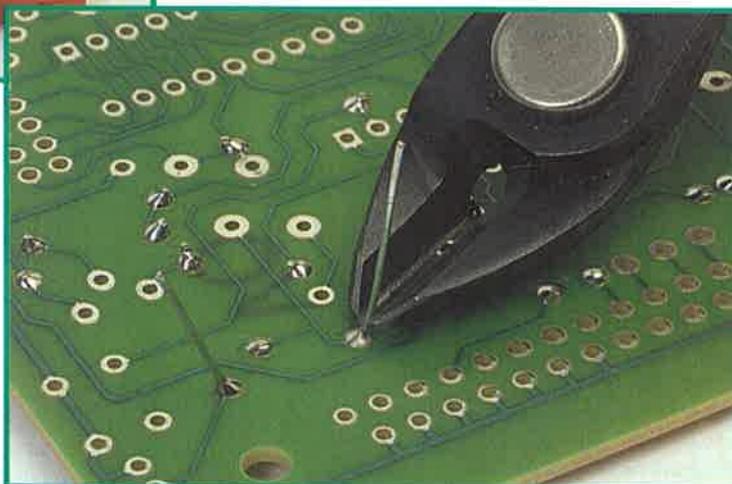
Il dissaldatore serve per eliminare lo stagno da una saldatura e poter estrarre il componente precedentemente saldato. Si basa sul movimento di una forte molla che produce il risucchio dello stagno fuso. Il processo di dissaldatura è molto più delicato e complesso rispetto a quello della saldatura. Per quanto possibile deve essere evitato.

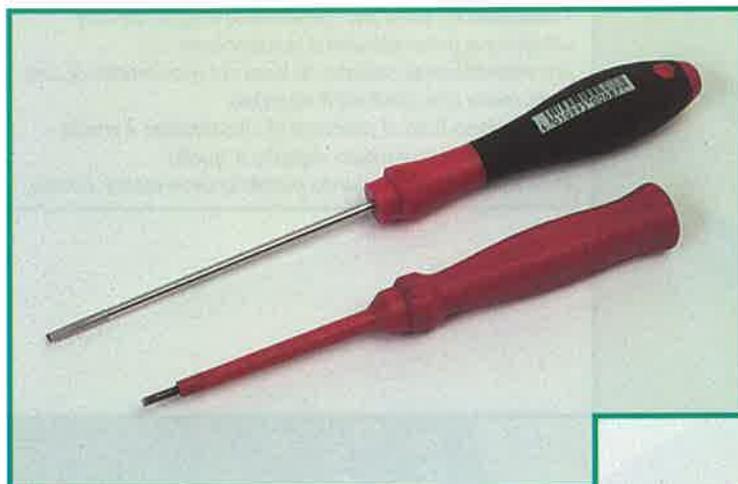
Per dissaldare scioglieremo lo stagno con il saldatore, quando questo sarà fuso, avvicineremo il dissaldatore alla saldatura, però dal lato opposto a quello del saldatore. In questo modo procederemo alla dissaldatura.



Il tronchesino è lo strumento adatto per tagliare i terminali dei componenti che sporgono oltre il cono di saldatura. Permette di tagliare i reofori grandi con precisione e senza problemi, e il risultato è migliore di quello che si avrebbe se si usassero delle forbici convenzionali.

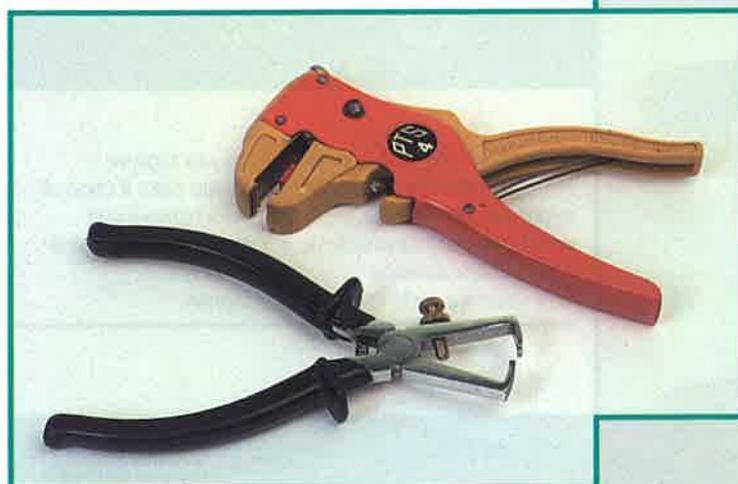
Il taglio del terminale sporgente deve realizzarsi leggermente al di sopra del cono ottenuto con la saldatura. Non si deve mai forzare il terminale, perché rischieremmo di sollevare la pista del circuito stampato.





Il cacciavite a taglio viene spesso usato in elettronica. Per esempio, è molto utile per estrarre i chips dagli zoccoli o per regolare i valori dei potenziometri.

Occasionalmente le forbici da elettricista possono servire per tagliare terminali, inoltre possiedono un incavo nella parte posteriore, che è utile per spelare i fili facilmente.



Se dobbiamo spelare molti cavi e di diverso diametro, lo strumento ideale è uno spelafili come quelli che si vedono nella figura, i quali permettono di realizzare tale operazione in modo molto rapido.

Inoltre è necessario avere delle piccole pinze, con cui si possono manipolare con precisione i componenti elettronici e far loro adottare la forma più appropriata.



MODULO DI CONTROLLO

PL 04

Montaggio passo a passo



Osservando la tabella dei codici delle resistenze potremo determinare il valore in Ω delle resistenze che dovremo saldare:

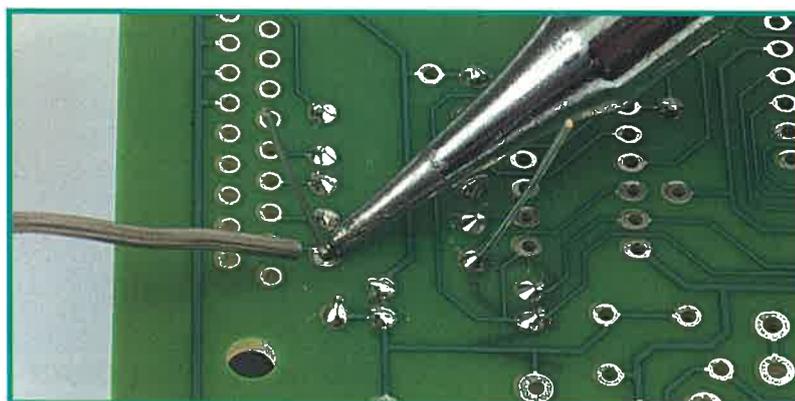
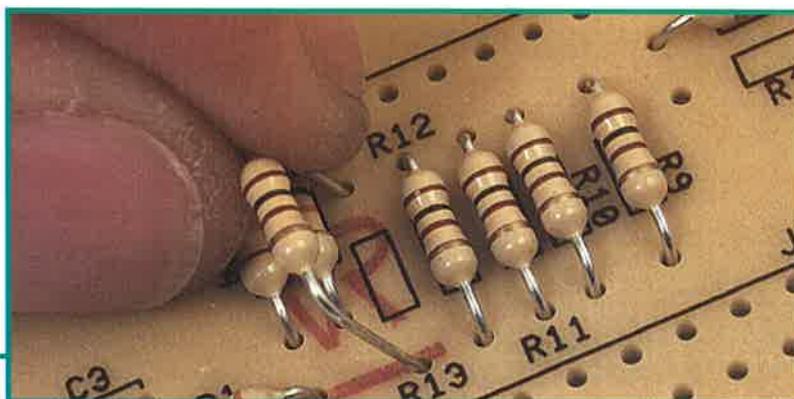
Marrone + nero + marrone: 100 Ω

Giallo + viola + rosso: 4,7 k Ω

Marrone + nero + rosso: 1 k Ω

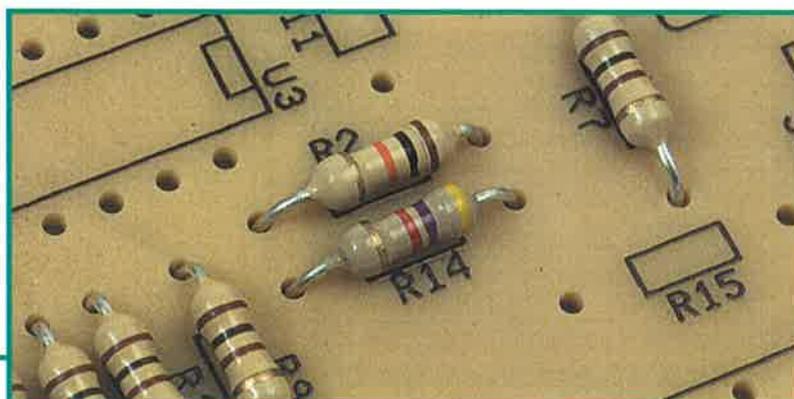
Ora identificheremo le resistenze da 100 Ω e le collegheremo in R11-R12 e R13.

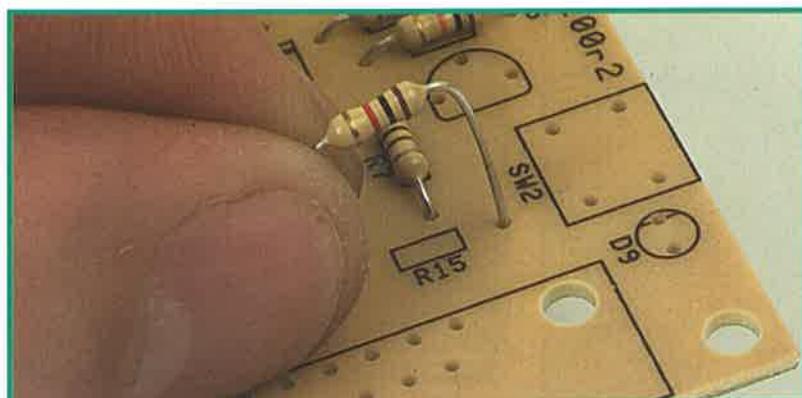
Le resistenze da 100 Ω da R9 a R13 serviranno per la protezione della porta parallela del PC.



Prima di saldare, riscaldaremo bene la piazzola sopra la quale fonderemo lo stagno. La saldatura deve rimanere a forma di cono, con lo stagno sufficiente per poter tappare il foro.

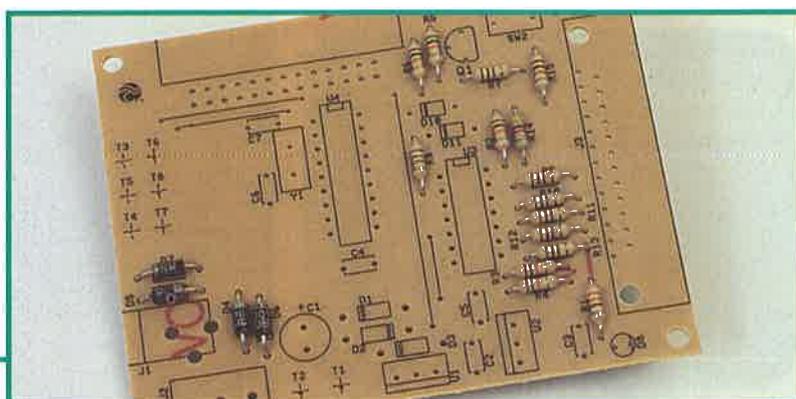
Ora identificheremo la resistenza da 4k7 Ω (equivalente a 4,7 k Ω , giallo-viola-rosso). La collegheremo e la salderemo nel posto assegnato sulla scheda di controllo per R14.





La resistenza da 1 k Ω va saldata in R15. Dopo aver collocato questa resistenza, avremo installato sulla scheda di controllo tutte le resistenze. A partire da adesso salderemo gli altri tipi di componenti.

Nella figura è indicato l'aspetto che deve avere il modulo di controllo a questo punto del montaggio. Dobbiamo assicurarci che tutte le resistenze e i diodi siano stati collocati nella loro posizione corretta, e che sia stata rispettata la polarità di questi ultimi.



Il motore che si vede nella foto è uno dei due che costituiranno il sistema di trazione di Monty. Questi motori serviranno per fornire trazione in avanti, in modo che attivando il motore di destra o quello di sinistra il robot girerà a destra o a sinistra. Sarà dotato di una riduzione e di un contenitore per essere inserito nello chassis.

Dovremo saldare due cavi al motore per poterlo collegare con l'elettronica di comando. Salderemo il cavo rosso (terminale positivo) al morsetto segnato con un punto. In realtà la polarità del motore è ininfluente, però il costruttore indica sempre un senso di rotazione convenzionale.



Consigli pratici



Il tester è uno strumento che permette di effettuare misure di tensione e di corrente, sia continua che alternata. Misura anche la conduttività, la resistenza dei componenti e, alcuni modelli, misurano la capacità dei condensatori e il guadagno dei transistor.

Le misure si eseguono per mezzo dei due puntali di prova. Quello di colore nero deve essere sempre collegato all'ingresso COM del tester, l'altro puntale è di colore rosso, e si collega nella posizione indicata sul tester a seconda che si debba misurare tensione, corrente, resistenza, eccetera.

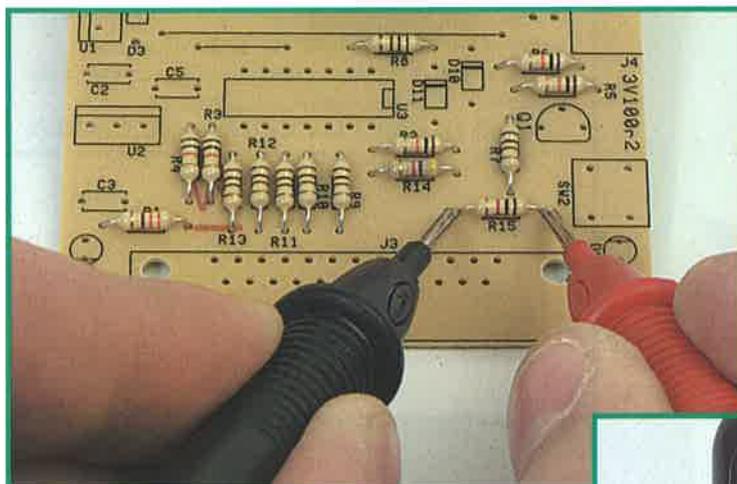


Mediante la manopola di selezione indichiamo al tester il tipo di misura che vogliamo realizzare. Il selettore è diviso in scale, e il numero che si vede a lato della scala, indica il valore massimo che può essere misurato.



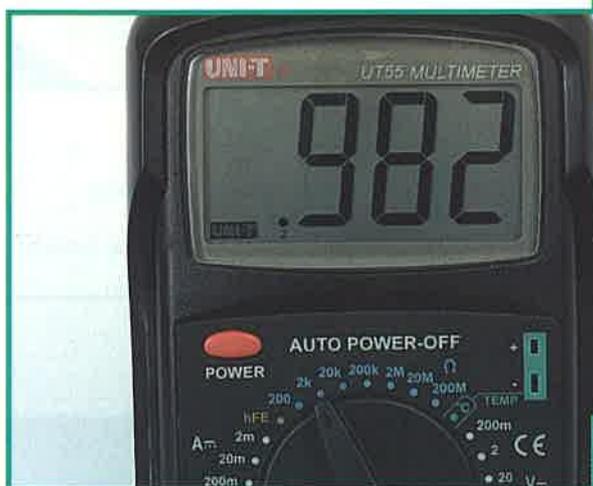
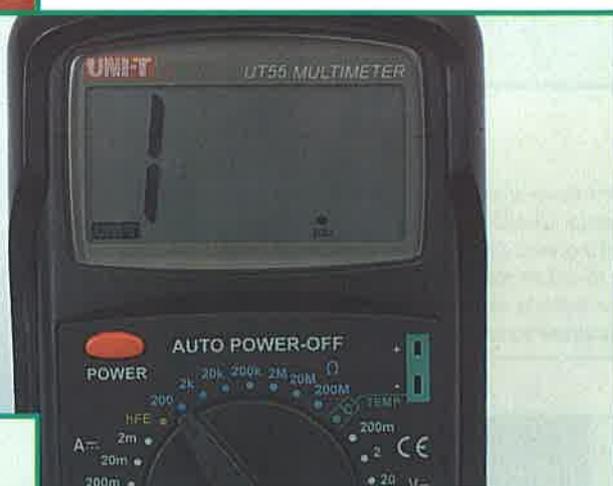
Per misurare il valore di una resistenza posizioneremo il selettore del tester nella zona contrassegnata dal simbolo Ω . All'interno di questa zona selezioneremo la scala più piccola.





Per misurare il valore di una resistenza dobbiamo posizionare i puntali in parallelo ad essa, e il circuito deve essere senza tensione. Proveremo a misurare le resistenze che abbiamo saldato sulla scheda, ma che non sono ancora collegate a nessun altro componente elettronico. Potremo notare una lieve differenza fra i valori reali e quelli teorici, che otterremo mediante la tavola del codice a colori.

Se sul display del tester appare il numero 1 a sinistra, significa che il valore della resistenza è superiore a quello che si può misurare con questa scala, quindi dovremo selezionare quella seguente, cambiando la posizione del selettore.



Quando potremo leggere sul display del tester un valore determinato, significa che ci troviamo in una scala valida. Il valore reale della resistenza sarà quello che ci viene indicato dal display in quell'istante, tenendo conto del fattore di moltiplicazione della scala in cui noi stiamo lavorando.

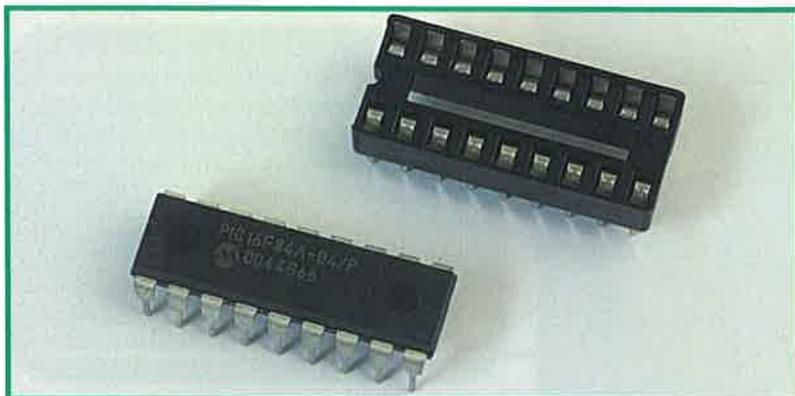
Potremo verificare la differenza esistente fra i valori reali delle resistenze, e quelli teorici che otterremo con la tavola del codice a colori. Lo scostamento deve essere contenuto all'interno del margine di errore segnalato dalla banda colorata riferita alla tolleranza.



MODULO DI CONTROLLO

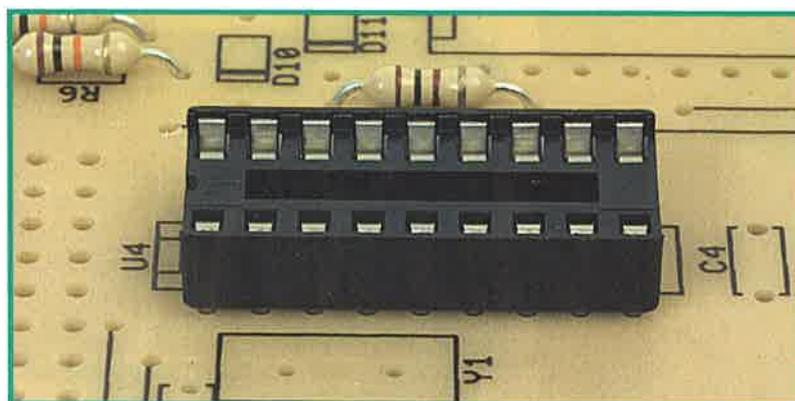
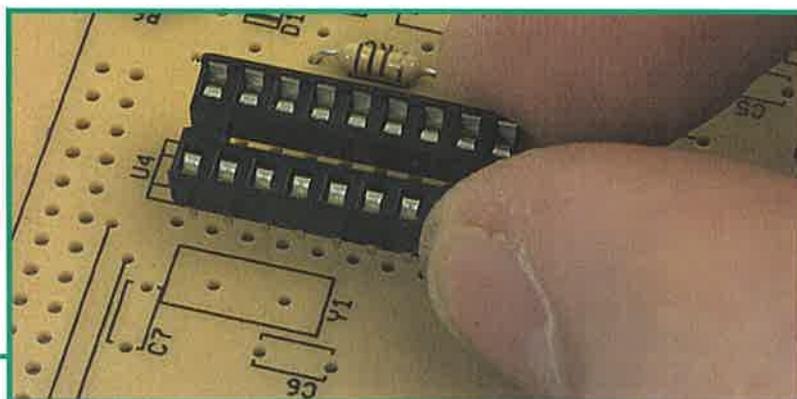
PL 06

Montaggio passo a passo



Salderemo uno zoccolo da 18 pin sulla scheda, sul quale monteremo successivamente il microprocessore PIC. Ogni volta che avremo bisogno di saldare uno zoccolo o un chip su una scheda, dovremo seguire il procedimento illustrato di seguito.

Lo zoccolo deve essere inserito sulla scheda, orientato secondo un preciso criterio. L'orientamento ci viene indicato da una piccola tacca di riferimento, che tutti gli zoccoli hanno su uno degli estremi. Sul circuito stampato questo orientamento ci viene indicato dalla serigrafia, su cui è riportato un disegno simile alla forma della tacca dello zoccolo.

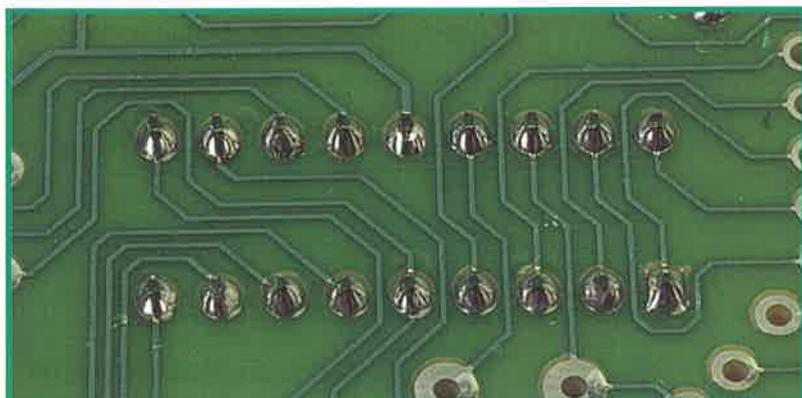


Lo zoccolo da 18 pin deve essere collocato sopra il simbolo U4 della serigrafia. Lo introdurremo sino a portarlo a filo della scheda, per permettere ai suoi piedini - che sono piuttosto corti - di fuoriuscire dall'altro lato, ed essere saldati correttamente.

Inizieremo saldando due pin agli angoli opposti della scheda, ad esempio potremmo saldare il pin in alto a destra e quello in basso a sinistra. Dopo aver eseguito queste due saldature, verificheremo che lo zoccolo sia posizionato a filo della scheda, e con l'orientamento corretto. Se si saldasse completamente e scopriremo poi un errore di montaggio, risulterebbe molto difficile estrarre lo zoccolo senza danneggiare le piste.

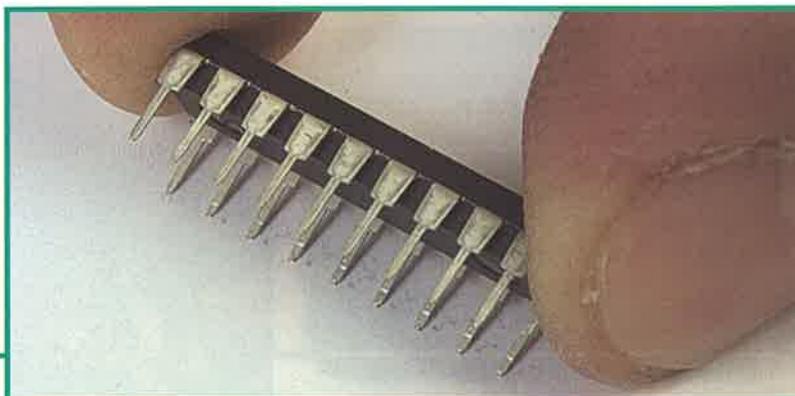


Montaggio passo a passo

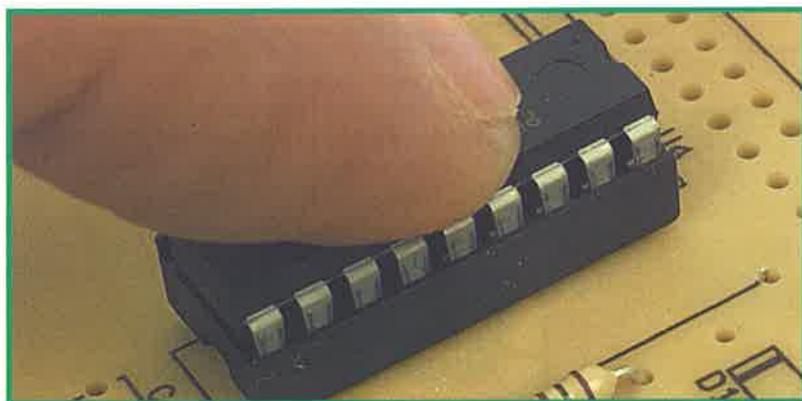


Dopo aver verificato la corretta posizione dello zoccolo procederemo alla saldatura dei rimanenti pin. Lo stagno attorno al pin dovrà assumere la forma di un cono, assicurando il contatto con la superficie, e non la forma di bolla. Non sarà necessario tagliare l'eccedenza dei pin, che spuntano dalla saldatura.

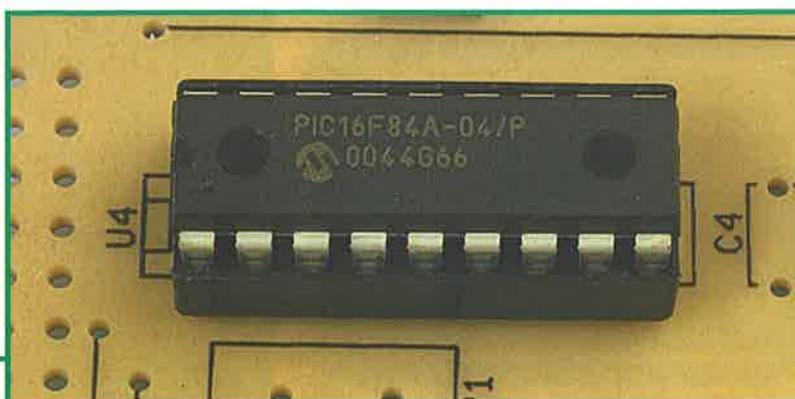
Tutti i chip, da nuovi, hanno i piedini leggermente aperti rispetto alla posizione che dovranno assumere quando saranno inseriti nello zoccolo. Per riallinearli nella posizione corretta (circa 90°) consigliamo di far pressione su di essi utilizzando una superficie liscia.



Introdurremo il PIC nello zoccolo con molta cautela, in modo da non danneggiare i piedini, facendo pressione sino a che non saranno completamente inseriti. La tacca del PIC deve coincidere con la tacca dello zoccolo, così come la tacca dello zoccolo deve coincidere con quella della serigrafia sulla scheda.



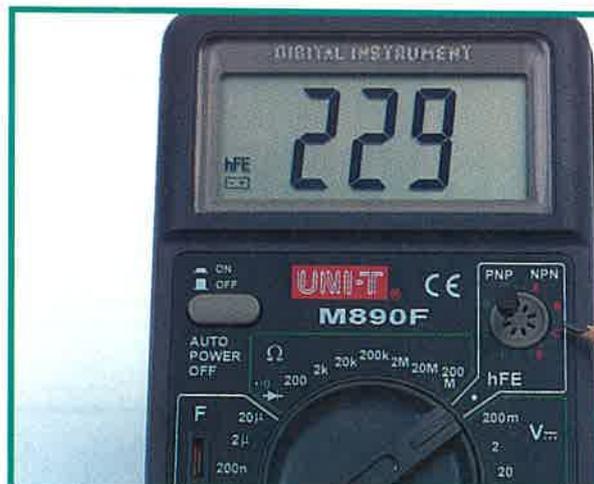
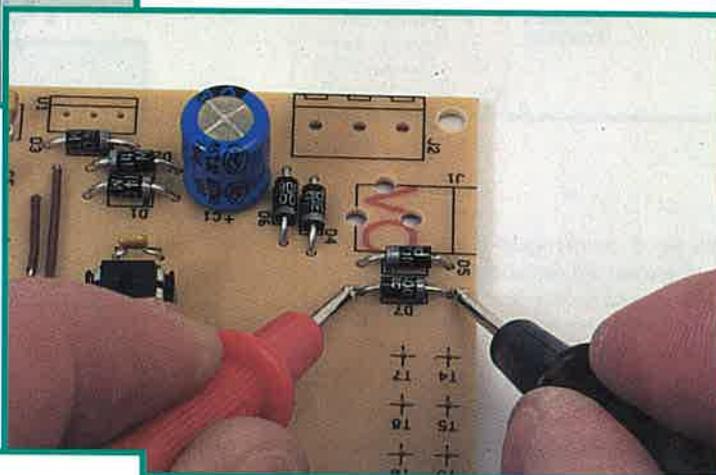
Il vantaggio di utilizzare uno zoccolo per montare il PIC sulla scheda, consiste nel fatto che, se avremo bisogno di toglierlo dalla scheda per sostituirlo con un altro, potremo farlo facilmente con l'aiuto di un cacciavite, evitando di doverlo dissaldare; operazione senza dubbio molto più complessa e delicata.





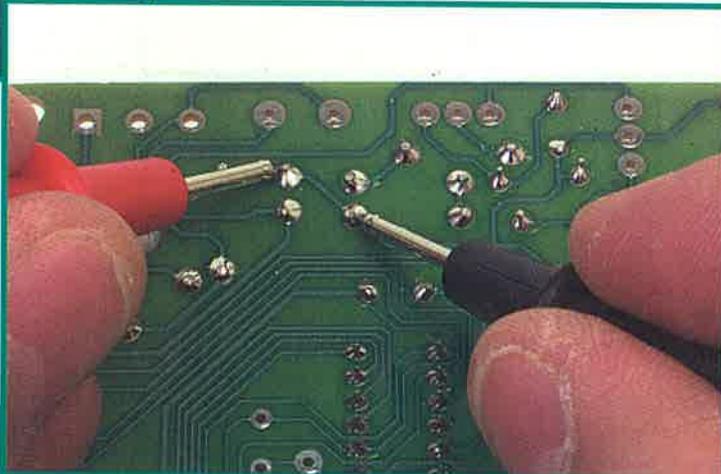
Una delle funzioni più utili che hanno alcuni tipi di tester, consiste nella misura dei diodi. Con questa funzione possiamo determinare lo stato interno di un diodo. Lo strumento misura la tensione di soglia del diodo quando questo viene polarizzato correttamente. Questa tensione di soglia è regolata a 0,3 V, per i diodi al germanio e a 0,7 V per quelli al silicio.

A titolo di esercizio, possiamo misurare lo stato di uno dei diodi che abbiamo saldato sulla scheda. I diodi sono elementi che permettono alla corrente di circolare in una sola direzione: nel verso che va dal catodo (banda grigia del diodo) verso l'anodo. In questo modo, se posizioniamo il puntale negativo del tester (colore nero) sul catodo e il puntale positivo sull'anodo del diodo, questo sarà polarizzato correttamente, e lo strumento segnalerà la tensione di soglia menzionata. Se scambiamo le punte del tester, il diodo resterà polarizzato inversamente, e lo strumento non darà una lettura valida.



Altri modelli hanno la capacità di verificare e misurare il guadagno dei transistor. Sulla ruota del selettore c'è una posizione indicata con hFE. Lo strumento dispone di uno zoccolo per l'inserimento del transistor. I piedini dello stesso, devono essere inseriti per il verso indicato nel tester. Sul video appare la lettura corrispondente al guadagno in montaggio a emettitore comune. Nel caso non si ottenga una lettura valida, il motivo è da ricercarsi nel transistor: non è stato correttamente inserito nello zoccolo, oppure è guasto.

Per ultimo, ricordiamo che molti strumenti hanno la capacità di misurare la continuità. Sulla selezione della scala c'è una posizione indicata con un altoparlante. In questa posizione lo strumento genera un piccolo suono ogni volta che si misura una resistenza di valore molto basso. Questa funzione ci permette di provare i fusibili, la continuità sulle piste, ecc. Per eseguire questa verifica, osserveremo la scheda dalla parte dove si trovano le piste saldate, e troveremo piste che uniscono i terminali dei componenti saldati. Se metteremo le punte di prova del tester sopra le saldature di questi componenti, e il tester non emetterà alcun suono, significa che avremo saldato male o che la pista è tagliata.



CODICI DEI CONDENSATORI

Le lettere usate come codici di tolleranza nei condensatori ceramici o a film plastico sono le seguenti:

C = +/-0.25pF	D = +/-0.5pF	F = +/-1%	G = +/-2%
J = +/-5%	K = +/-10%	M = +/-20%	Z = -20 +80%

I colori indicano i coefficienti termici elettrici e sono i seguenti:

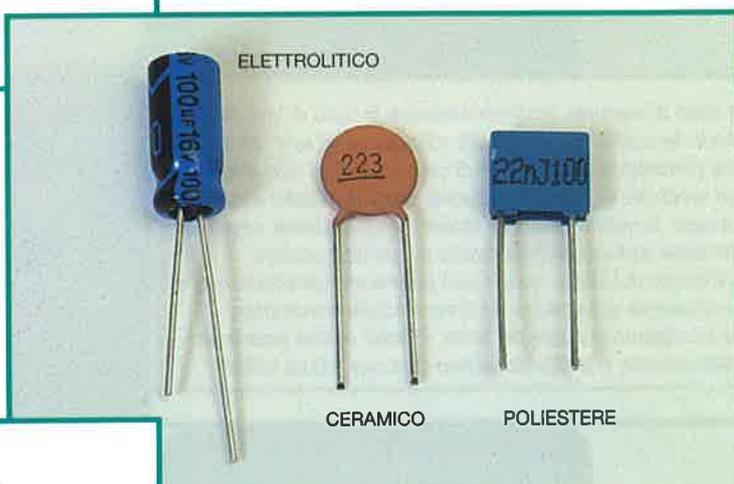
verde = high K	giallo = medium K	rosso/viola = +100 ppm/deg C
nero = zero TC	marone = -150 ppm/deg C	viola = -750 ppm/deg C
marone/rosso = -1500 ppm/deg C		

Il codice delle lettere usate per indicare il dielettrico:

MKT = Metallised polyester
MKC = Metallised polycarbonate
KT = Polyester film/foil
KS = Polystyrene film/foil
KP = Polypropylene film/foil
MKP = Metallised polypropylene

Nella tabella sono illustrati i codici impiegati normalmente nella descrizione dei condensatori: la loro tolleranza (livello di errore), il loro coefficiente termico dielettrico, o il tipo di dielettrico che utilizzano. Alcuni condensatori hanno un codice a colori per determinare il valore. Viene impiegato lo stesso codice delle resistenze. Gli altri - la maggioranza - hanno impresso un numero, che indica direttamente la capacità dei condensatori.

I tre tipi di condensatori più comuni sono: i ceramici, in poliestere, ed elettrolitici. Questi ultimi sono quelli che raggiungono le maggiori capacità.

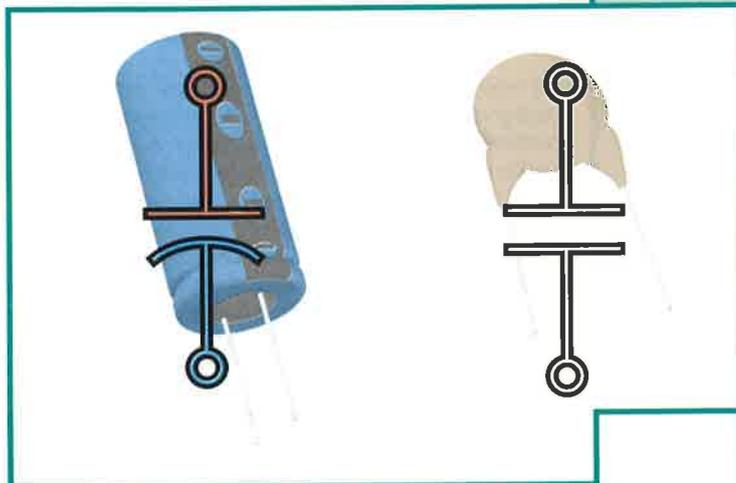


ELETTROLITICO

CERAMICO

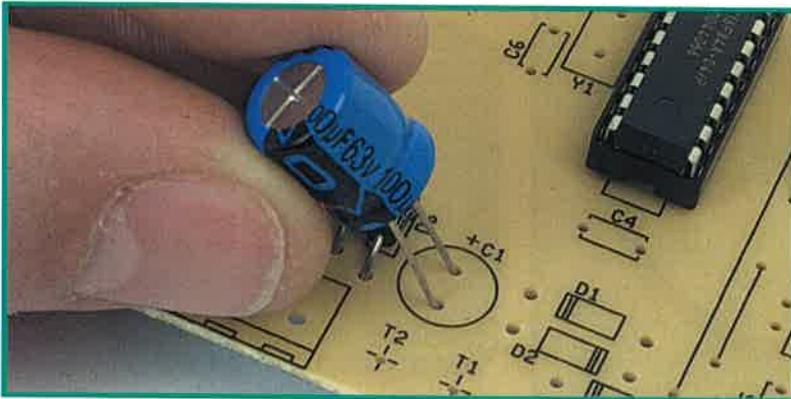
POLIESTERE

I condensatori elettrolitici sono gli unici che possiedono polarità, questa viene indicata negli schemi elettrici mediante un simbolo. Per il resto dei condensatori, dato che non possiedono polarità, è indifferente il verso con cui vengono montati. Pertanto nel simbolo elettrico di questi elementi non viene indicata nessuna polarità.



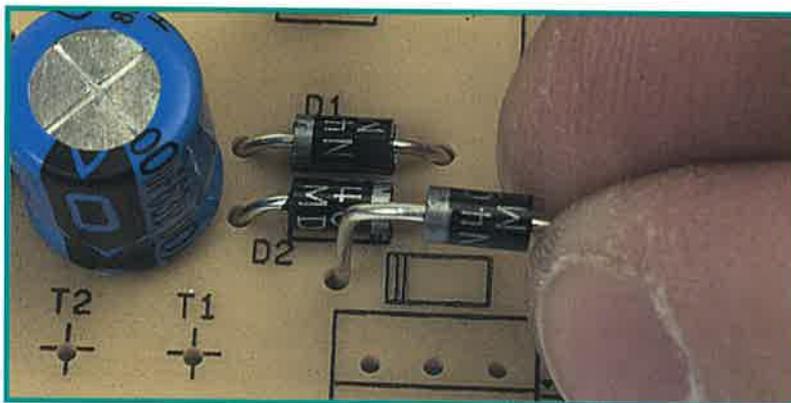
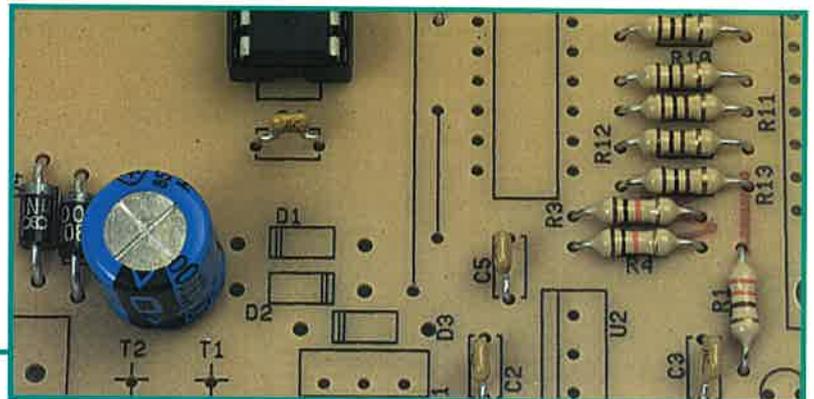
Il capacimetro è uno strumento che permette di misurare i valori dei condensatori. Alcuni modelli di multimetri o tester includono un capacimetro. Nella manopola di selezione, alcune scale indicate con F, permettono di selezionare il valore massimo del condensatore che si vuole misurare. I piedini del medesimo si inseriscono nello zoccolo appropriato dello strumento.





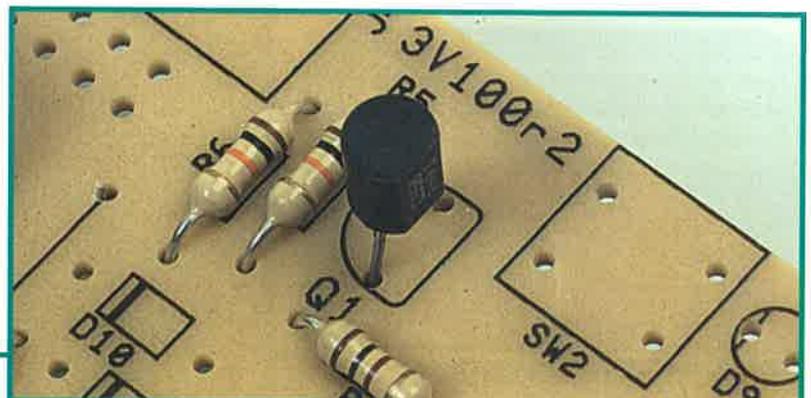
Prima collocheremo il condensatore elettrolitico da 100 μ F/63 V in C1. Questo condensatore ha polarità. Il lato negativo viene indicato sul condensatore con una banda laterale con il segno "-". Il piedino negativo, inoltre, è il più corto dei due. Sulla serigrafia viene segnato il piedino positivo del condensatore con il segno "+". Deve essere orientato come indicato nella figura.

Adesso collocheremo quattro condensatori ceramici da 100 nF. Questi condensatori non hanno polarità, quindi è indifferente il senso in cui verranno inseriti. Vanno collocati su C2-C3-C4 e C5.



Montiamo ora i tre diodi rettificatori 1N4007. Questi diodi hanno polarità, essendo il catodo (terminale negativo) segnato con una banda grigia. Dovremo montarli in D1-D2 e D3, facendo coincidere la banda grigia del diodo con la riga nera disegnata sulla serigrafia, come abbiamo già fatto per i diodi da D4 a D7.

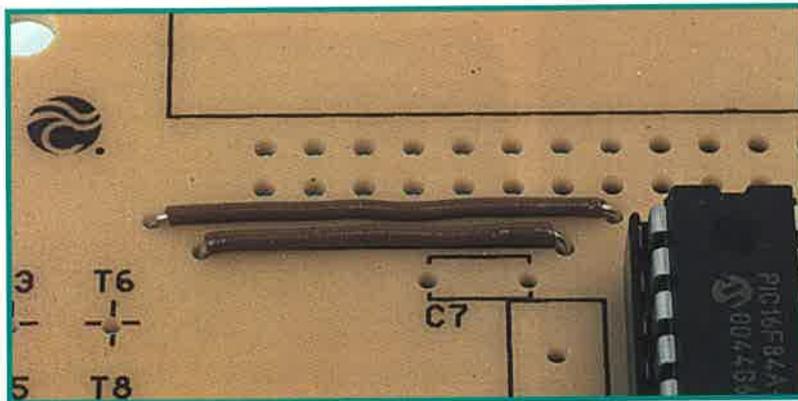
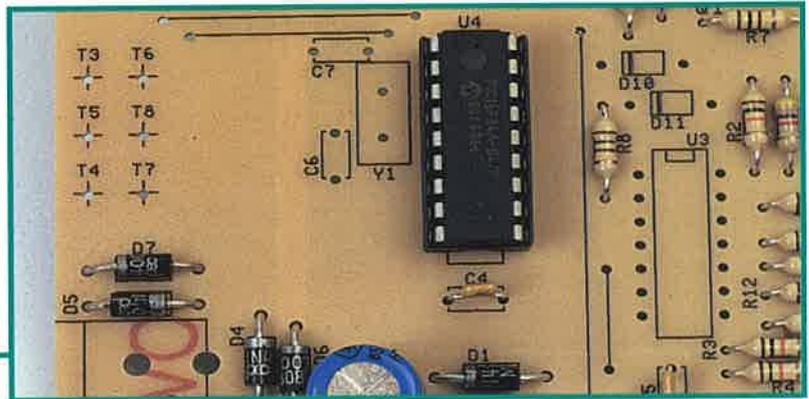
L'elemento seguente da saldare è il transistor modello BC558, che va montato in Q1. Dobbiamo inserire il transistor in modo che, visto dall'alto, coincida con la forma a "mezzaluna" disegnata sulla serigrafia. Il transistor non deve essere montato a filo della scheda, ma posto ad una altezza di circa 5 mm rispetto ad essa.





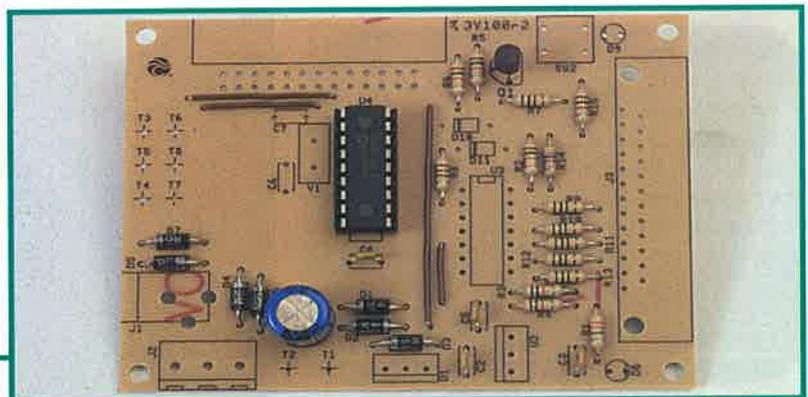
La scheda di controllo è un circuito stampato a singola faccia, dato che contiene le piste solo sulla parte inferiore e non su entrambi i lati. Sarà necessario realizzare alcuni ponticelli sulla parte superiore della scheda per poter collegare le piste che non abbiamo unite nella parte inferiore. Per i ponticelli utilizzeremo questo cavo.

In totale dovremo realizzare quattro ponticelli. Sulla serigrafia, vengono indicati mediante quattro linee nere che uniscono i due punti corrispondenti, come mostra l'immagine.



Dovremo tagliare il cavo nella misura appropriata per ognuno dei ponticelli. Una volta tagliato lo introdurremo dalla parte superiore della scheda, saldando i terminali che fuoriescono dalla parte inferiore. Dobbiamo montarlo a filo della scheda come viene mostrato nell'immagine.

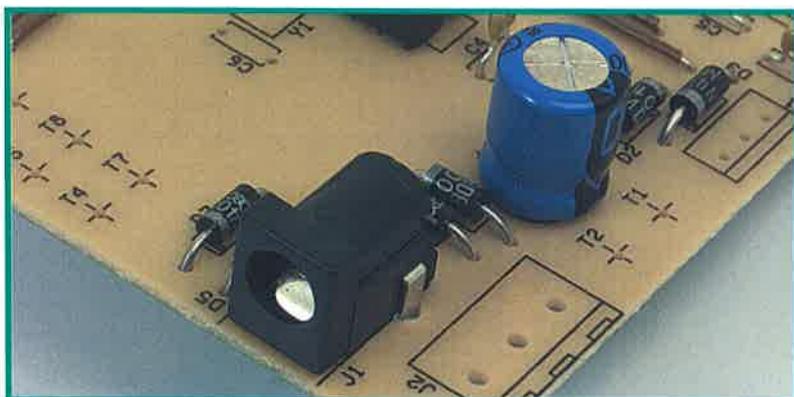
Questo è l'aspetto della scheda di controllo in questo momento. Abbiamo già saldato i quattro ponticelli, tutte le resistenze e la maggior parte dei diodi e dei condensatori.



MODULO DI CONTROLLO

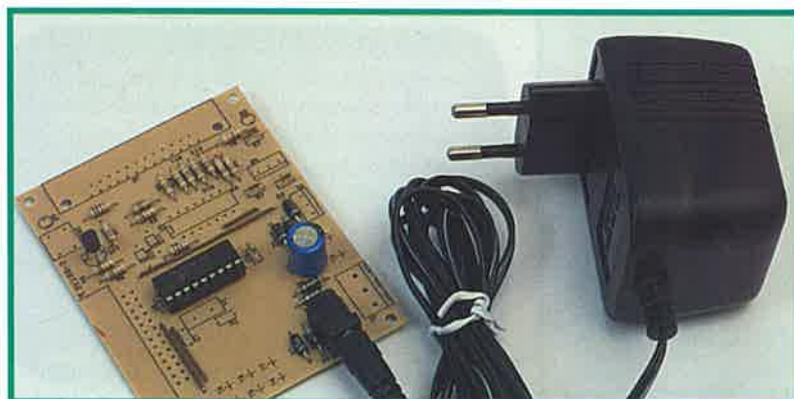
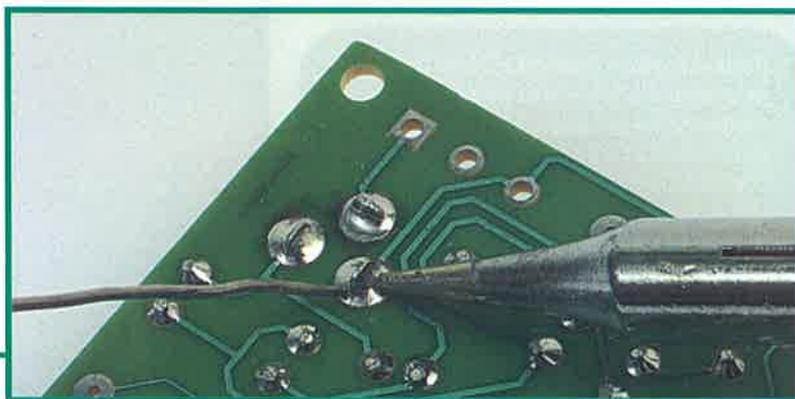
PL 09

Montaggio passo a passo



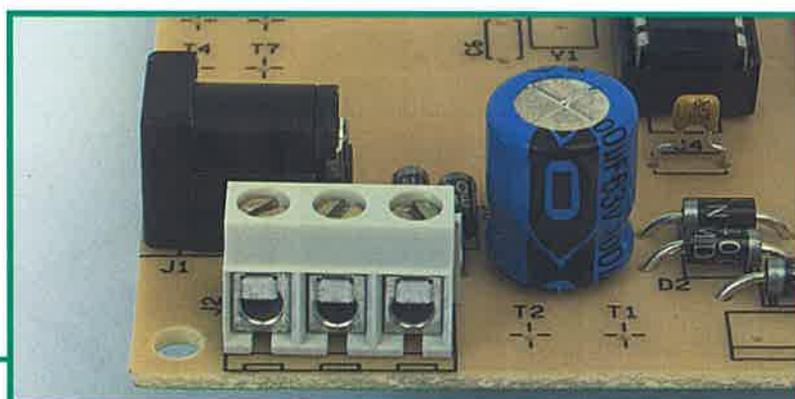
Prima salderemo il connettore di alimentazione per il circuito stampato. Questo va inserito in J1, dobbiamo fare pressione su di esso fino a che sia totalmente introdotto e a filo della scheda.

Dobbiamo saldare i piedini del connettore di alimentazione con sufficiente stagno, sciogliendolo rapidamente fino a che ricopra tutta la superficie della piazzola, per poi lasciarlo raffreddare. Se realizziamo questa saldatura molto lentamente, corriamo il rischio che lo stagno salga su per il piedino del connettore e passi dall'altro lato della scheda.



Il connettore di alimentazione è predisposto per collegare un trasformatore che alimenti la scheda di controllo. Il trasformatore deve essere di 12 VA (corrente alternata), come quello che è mostrato nella figura.

Ora monteremo la morsettiere a tre contatti. Attraverso questa morsettiere potremo alimentare Monty con la corrente continua prodotta dalle pile che inseriremo nell'apposita sede. Dovremo collocare le morsettiere orientate verso l'esterno della scheda, come illustrato nella fotografia.

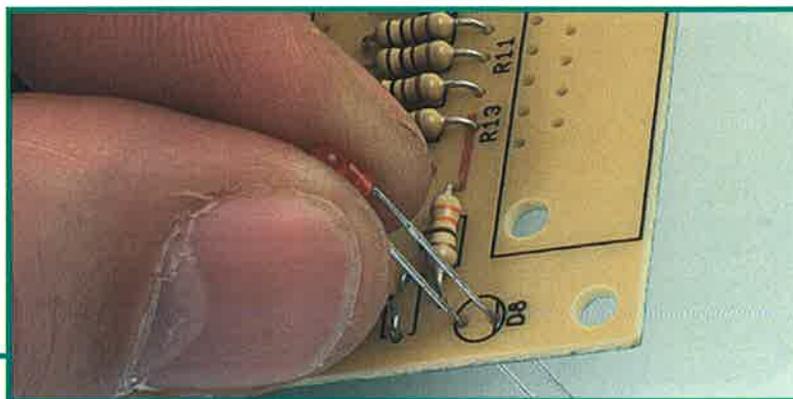


Montaggio passo a passo

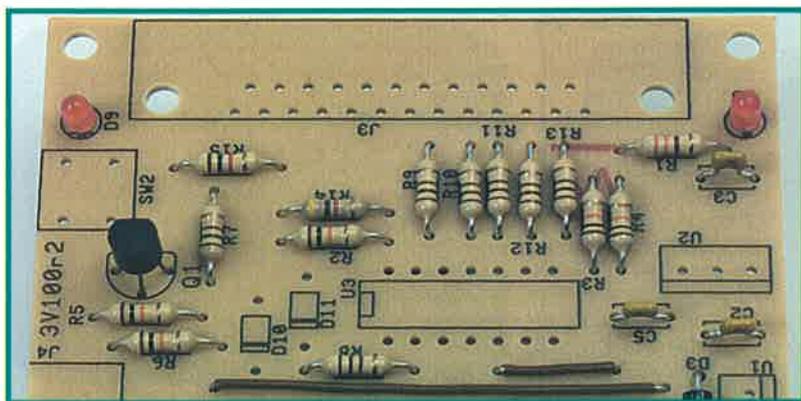


Il LED è un diodo di tipo speciale, che si illumina quando circola corrente attraverso di esso. Questo diodo, come tutti gli altri, ha una polarità. Il catodo (terminale negativo), è il piedino più corto. Si può anche individuare quale sia il catodo guardando il contenitore, dato che ha una zona leggermente rifilata.

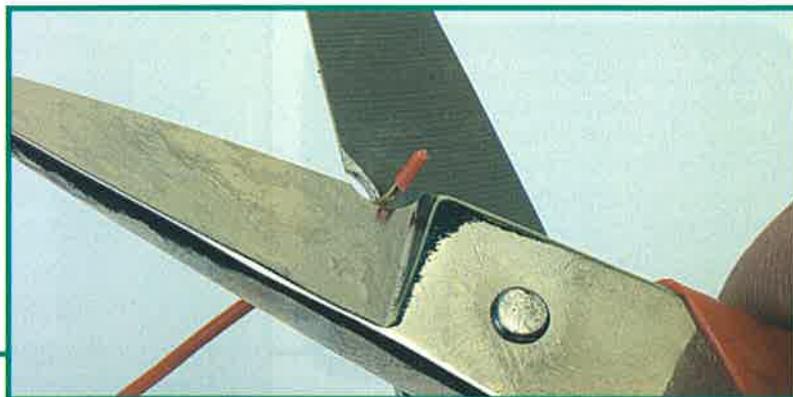
Dobbiamo saldare due diodi LED in D9 e D8. Nella serigrafia della scheda troveremo un cerchio che indica il diodo LED. In questo cerchio troveremo una linea nera in una delle parti laterali che rappresenta la tacca che possiede il LED, e che segnala il catodo. Dobbiamo far coincidere questa linea nera con il catodo del diodo.



Ecco come devono risultare i due diodi LED saldati: perpendicolari rispetto alla scheda e ben dritti. Dobbiamo evitare, per questioni estetiche, che restino inclinati, dato che i diodi LED sono componenti elettronici molto vistosi.



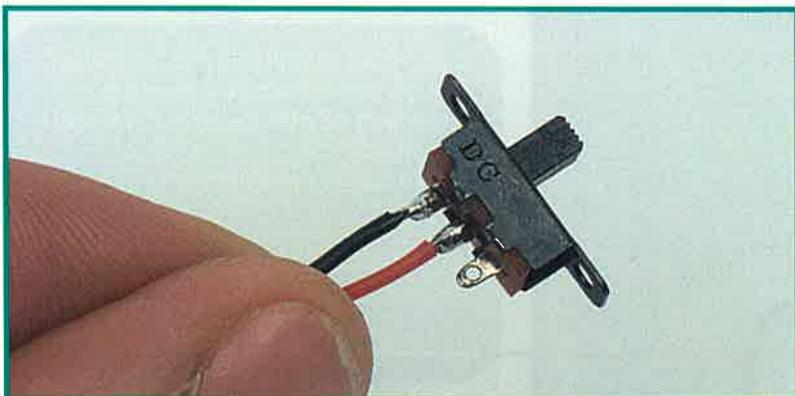
Ora passeremo a spelare le estremità dei cavi per il commutatore. Per fare ciò, useremo uno spelafili, o delle forbici. Nel caso usassimo le forbici, queste sono dotate di un incavo nella parte interna che facilita la spelatura del cavo.



MODULO DI CONTROLLO

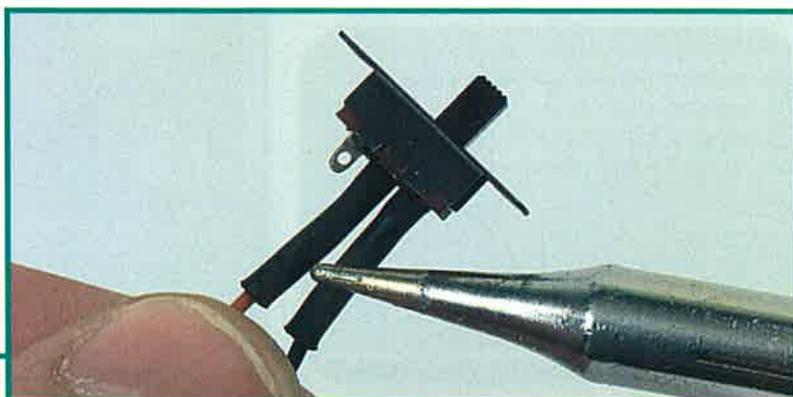
PL 10

Montaggio passo a passo



Ora, introduciamo un estremo dei due cavi nei fori dei piedini dell'interruttore. Un filo nel foro centrale e un altro in uno dei due estremi. Saldiamo ora i fili all'interruttore.

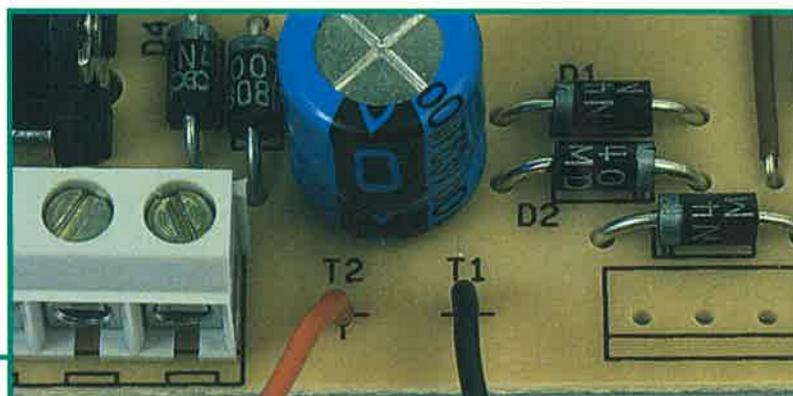
Dopo aver saldato i due cavi all'interruttore, dovremo collocare il termorestringente tagliato in due parti uguali di circa 1,5 cm. Questo è un isolante la cui funzione è quella di evitare che le due parti metalliche dei piedini dell'interruttore entrino in contatto, provocando corto circuito.



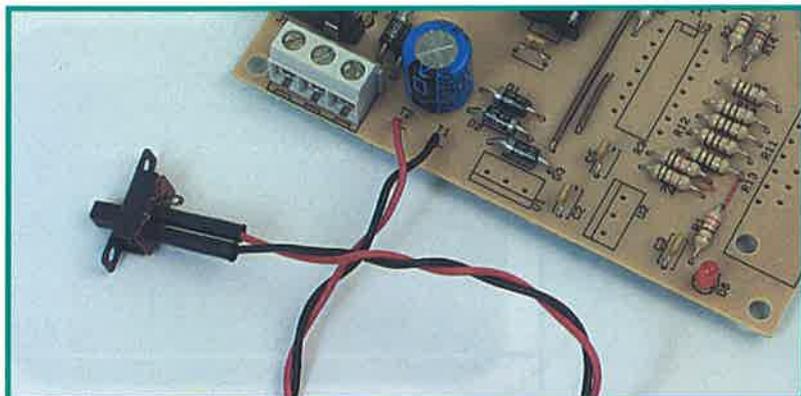
Il termorestringente si introduce in ognuno dei due piedini utilizzati, quindi si riscalda con il saldatore perché si restringa e aderisca totalmente ai piedini e al cavo. Il risultato finale deve essere simile a quello della fotografia.



Adesso salderemo l'interruttore alla scheda di controllo. Per questo introdurremo gli altri due estremi dei cavi, già spelati, nei terminali T1 e T2 di SW1 nella parte superiore della scheda. Salderemo il pezzo di cavo spelato che fuoriesce dalla parte inferiore.



Montaggio passo a passo



Questo è l'aspetto che deve avere l'interruttore SW1, che sarà l'interruttore di accensione e spegnimento del robot, permettendo che passi la corrente dalla fonte di alimentazione a quella elettronica.

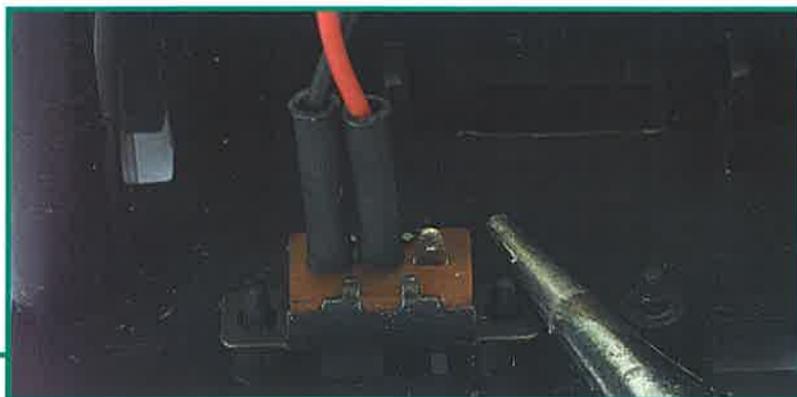
L'interruttore andrà inserito nel telaio. Il telaio dispone di due fori nella parte inferiore, nei quali inseriremo due interruttori. Introduremo SW1 nel foro più piccolo.

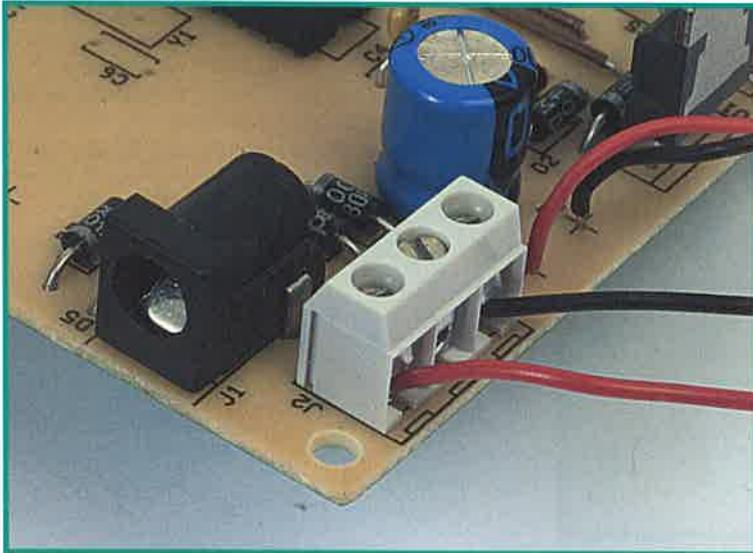


Per fare in modo che l'azionamento dell'interruttore corrisponda con i simboli ON/OFF stampati sulla base, dovremo introdurlo come mostra l'immagine. La leva mobile dell'interruttore fuoriuscirà dalla parte inferiore del telaio.



Se vogliamo fissare l'interruttore in questa posizione, riscaldiamo col saldatore i due puntelli di plastica in cui l'avremo inserito. Questa plastica si scioglierà e fisserà l'interruttore. Vi raccomandiamo di fare questa operazione solo più avanti, quando la scheda di controllo sarà completamente montata.

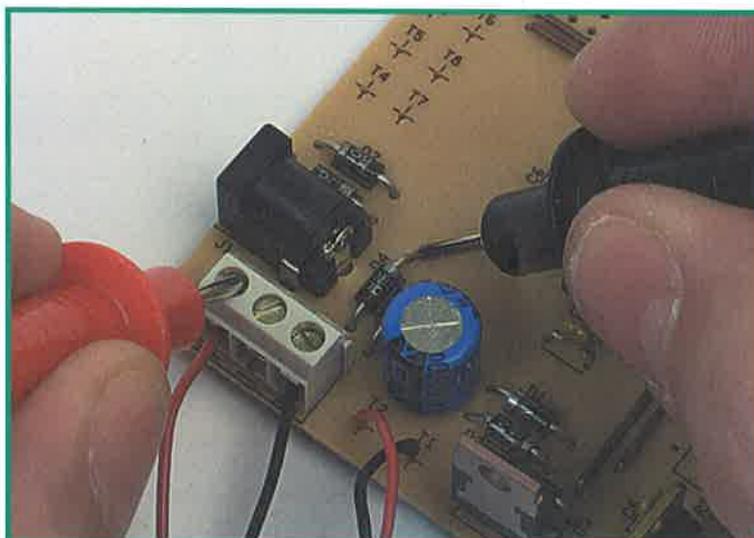
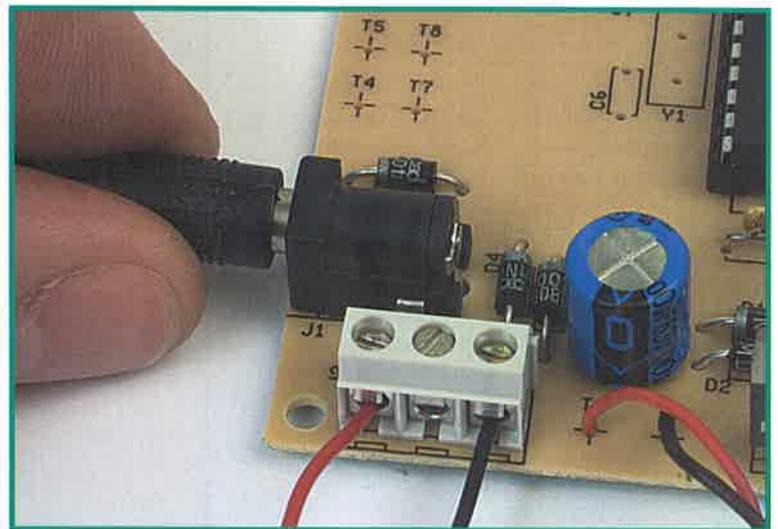




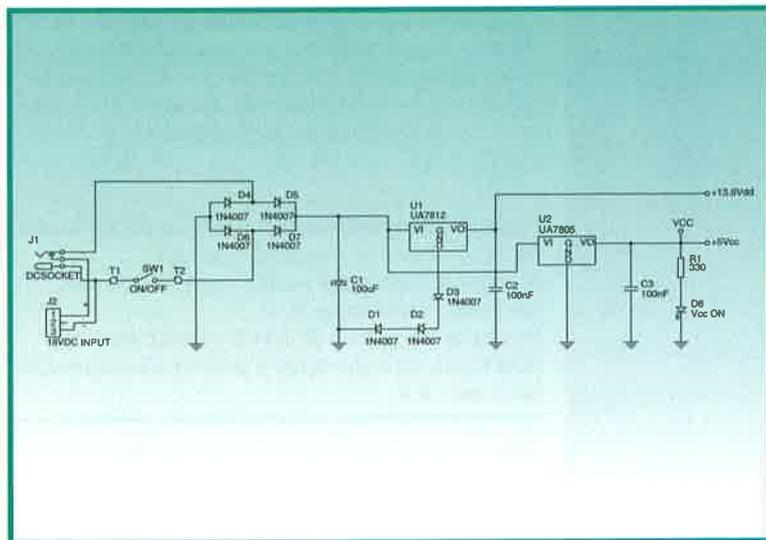
Cominceremo dall'analisi dell'alimentatore di Monty. Esso si trova sulla scheda di controllo, ed ha il compito di fornire 5 V per alimentare la circuiteria e 13.8 V necessari durante il processo di scrittura del microprocessore.

Possiamo alimentare Monty tramite un trasformatore da 12 VA (corrente alternata) o con 9 V di corrente continua. Il trasformatore è collegato mediante il connettore J1, mentre la morsettiera J2, con la polarità mostrata nella figura, cioè con la parte positiva alla sinistra che serve per i 9 V.

La scheda di controllo è stata progettata in modo che non sia possibile alimentarla contemporaneamente con una pila da 9 V tramite J2 e con un trasformatore tramite J1. Per questo quando inseriamo il connettore del trasformatore in J1, viene automaticamente interrotta l'alimentazione che arriva da J2.

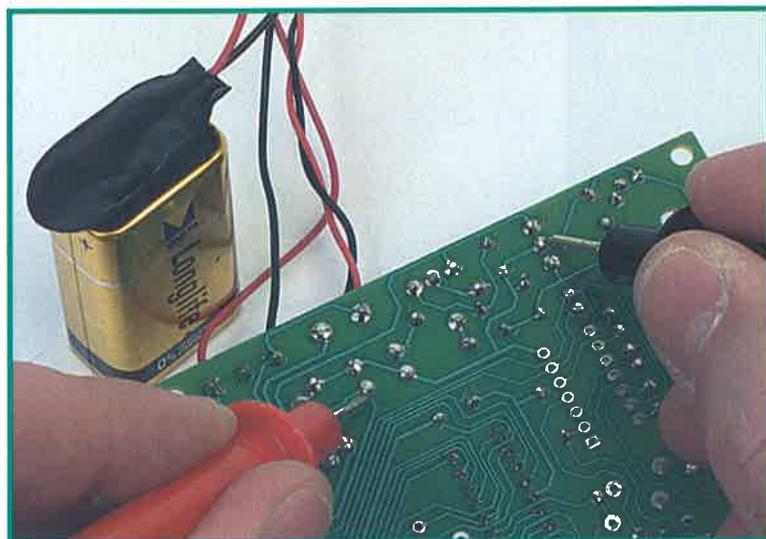
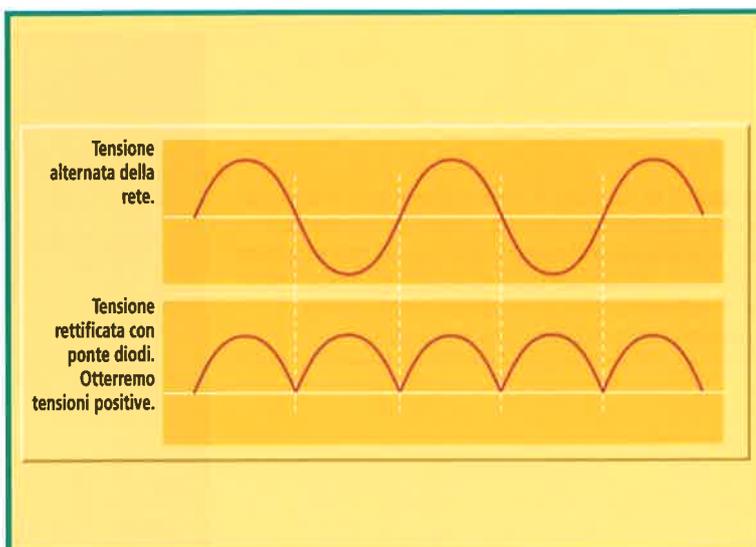


Misura di continuità. Possiamo verificare quanto abbiamo detto tramite un tester; se J1 è libero c'è conduttività fra i terminali mostrati in figura, altrimenti no. Se alimentiamo il circuito con tensione alternata, è necessario rettificare e filtrare il segnale per ottenere una tensione continua. Nel caso che la tensione di ingresso sia continua questi due stadi non sono necessari, e si passa direttamente allo stadio di stabilizzazione della tensione.



Schema dello stadio di rettificazione e filtraggio. Lo stadio di rettificazione si realizza mediante un ponte a diodi formato da D4-D5-D6 e D7. Questo stadio trasforma la tensione di ingresso, che ha valori positivi e negativi, in un segnale che contiene solo valori positivi, come mostrato nella figura sotto.

Rettifica della tensione alternata con il ponte diodi. La tensione positiva presente sull'uscita del ponte diodi è pulsante e non ancora adatta per alimentare i circuiti. Per questo è necessario realizzare uno stadio di filtraggio con l'aiuto di un condensatore, che abbia un ciclo di carica e scarica tale per cui si possa ottenere una tensione continua di valore costante all'uscita. Ora dovremo solo regolare e stabilizzare questa tensione entro le soglie di alimentazione del nostro circuito.

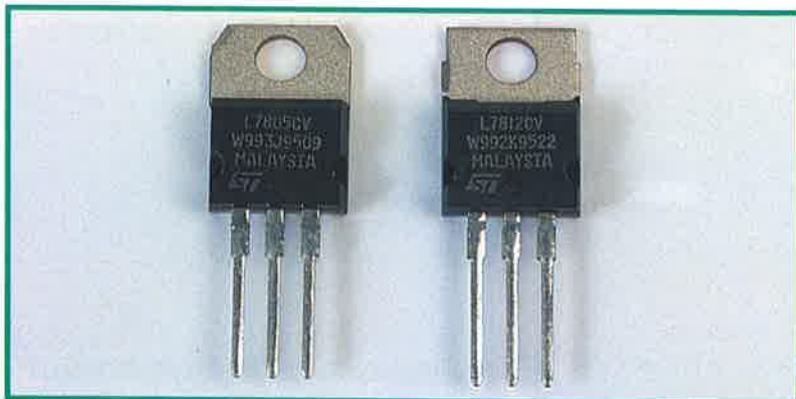


Misura di tensione fra lo stadio di filtraggio. Possiamo misurare il valore della tensione positiva presente sull'uscita dello stadio di filtro. Per questo alimenteremo la scheda con un trasformatore da 12 VA, e regoleremo il tester in misura di corrente continua sulla scala dei 20 V. Posizioneremo i terminali come mostrato in figura, e otterremo una tensione di 15 V.

MODULO DI CONTROLLO

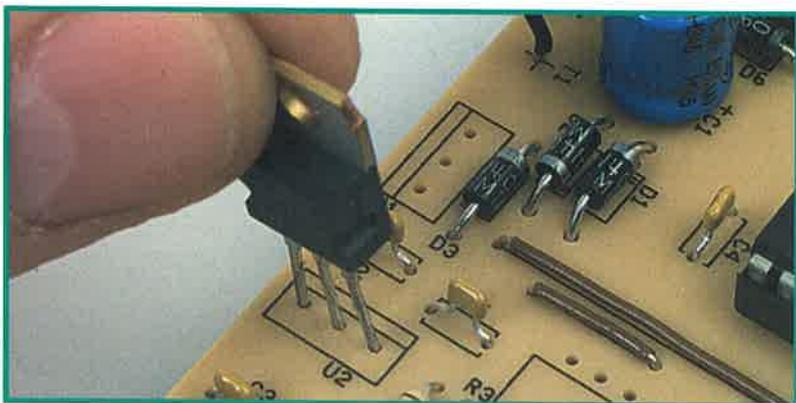
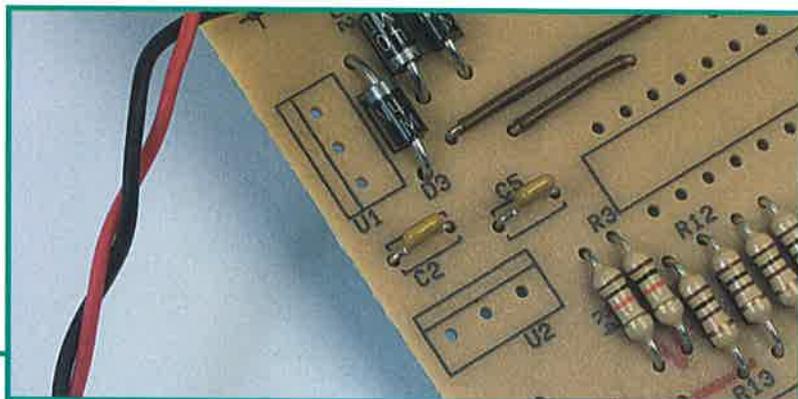
PL 12

Montaggio passo a passo



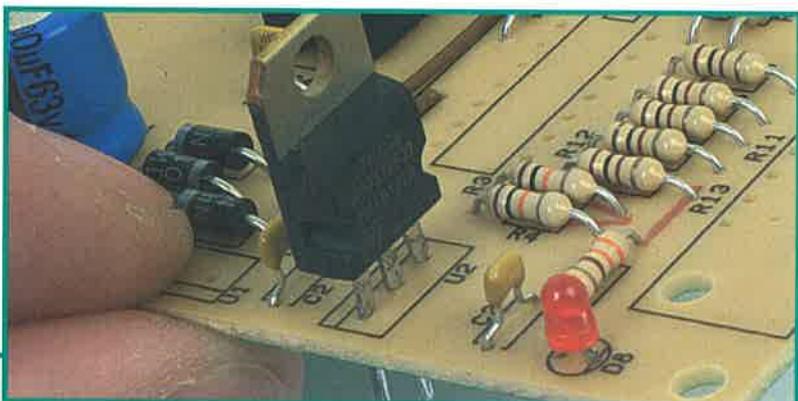
Montiamo ora i due regolatori che serviranno per stabilizzare la tensione. Il componente 7805 fornisce una tensione di 5 V per i circuiti elettronici, mentre il 7812 unito ai tre diodi D1 D2 e D3 fornisce i 13.8 V necessari per la programmazione del PIC.

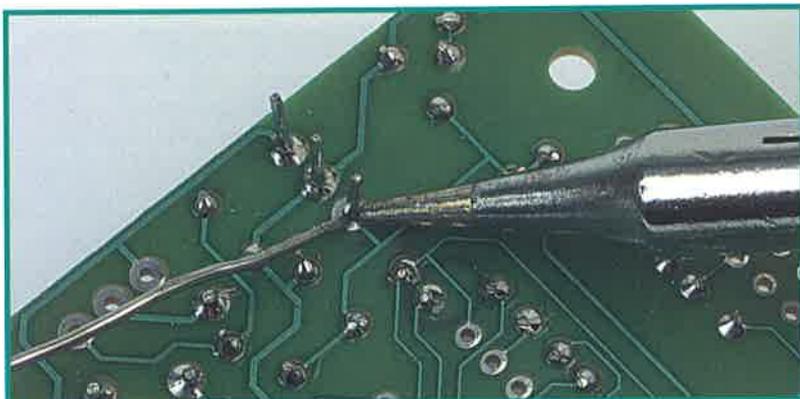
È molto importante posizionare questi due regolatori nella posizione corretta e nel posto che ad essi compete. Il regolatore 7812 andrà collocato in U1 e il 7805 in U2. La serigrafia ci indica con una doppia linea nera il verso di inserzione dei regolatori.



Guardando la scheda dall'alto osserviamo che posizionando correttamente il regolatore, ad esempio il 7805, la vista del componente coincide con quella mostrata nella serigrafia: la doppia linea nera coincide con la parte metallica brillante del regolatore.

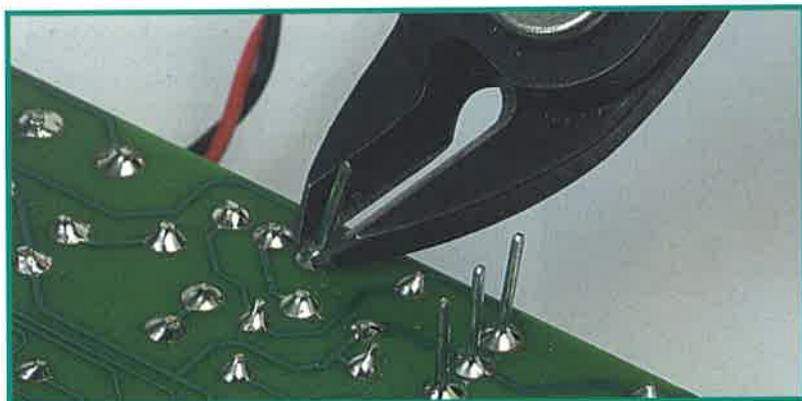
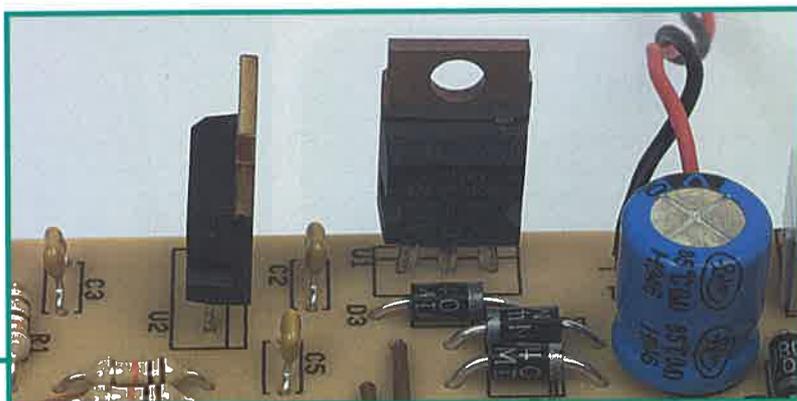
Dobbiamo introdurre il regolatore sino al punto permesso dai suoi piedini. Procederemo ora a saldare il 7805 montato su U2; prima di eseguire la saldatura ci assicureremo di averlo montato con l'orientamento mostrato in figura.





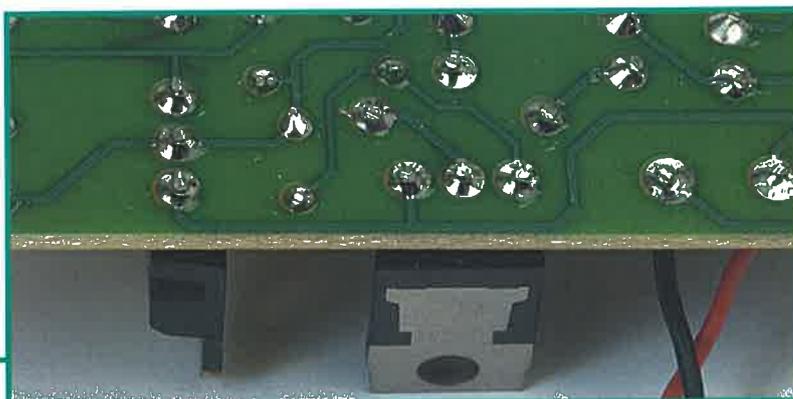
Realizzeremo le saldature di questi due regolatori con speciale attenzione, dato che attraverso di essi circolerà tutta la corrente che alimenterà la nostra scheda. Dovremo riempire con lo stagno tutta la piazzola di saldatura.

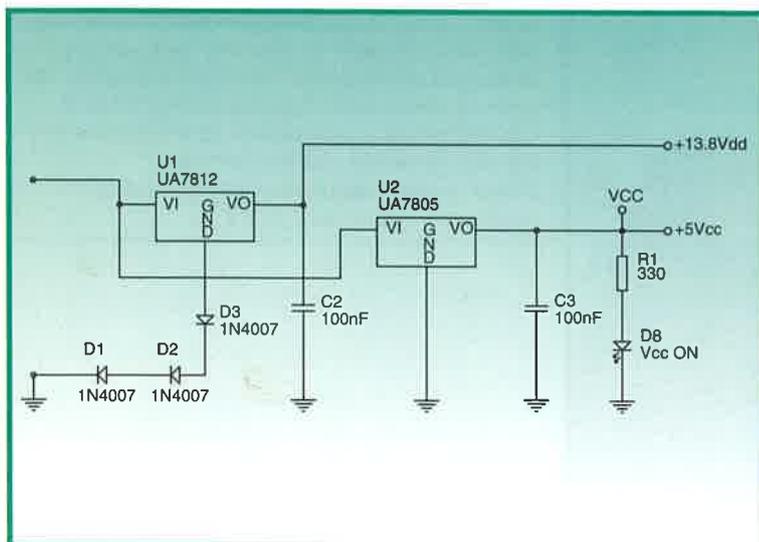
Ora monteremo il 7812 nella posizione U1 della scheda, secondo il verso mostrato nella figura: dritto e totalmente inserito.



Una volta realizzate le saldature di questo regolatore, procederemo a tagliare la parte eccedente dei piedini. Dato che i piedini di questi componenti sono più resistenti di quelli delle resistenze o dei diodi, raccomandiamo di utilizzare un tronchesino.

Questo è l'aspetto finale che devono avere le saldature dei regolatori. È importante che brillino e che formino un cono appoggiato totalmente sulla superficie della scheda. Ricontrolleremo nuovamente che i regolatori siano stati inseriti nelle posizioni indicate, dato che un errore in questa sezione potrebbe creare dei gravi danni alla scheda.





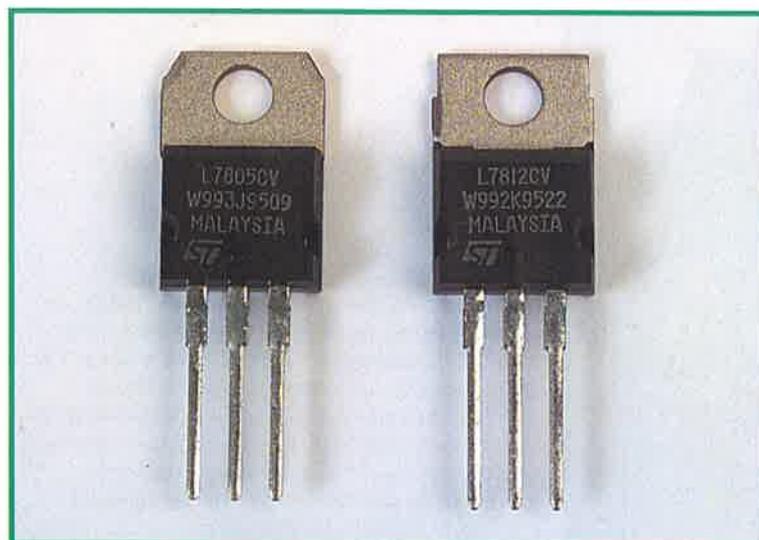
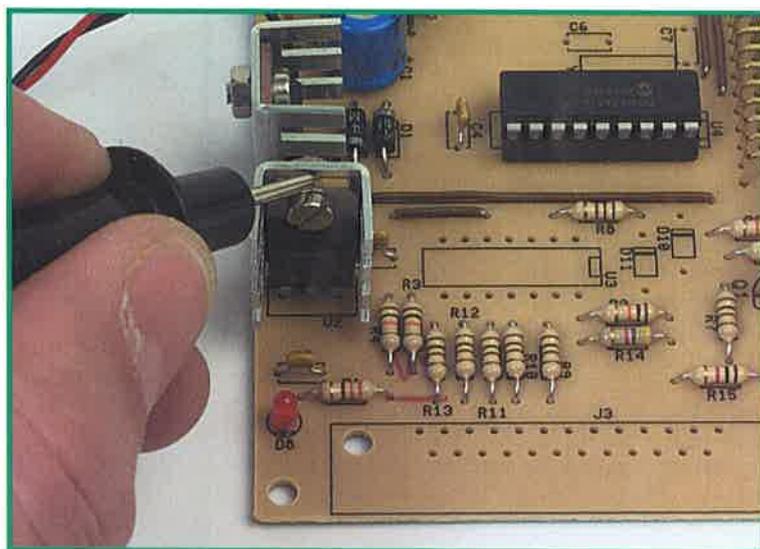
Una volta che avremo la tensione filtrata all'uscita del condensatore C1, dovremo stabilizzarla per ottenere i 5 V necessari per alimentare i circuiti

e i componenti elettronici della scheda, e i 13,8 V necessari durante il processo di programmazione del Pic.

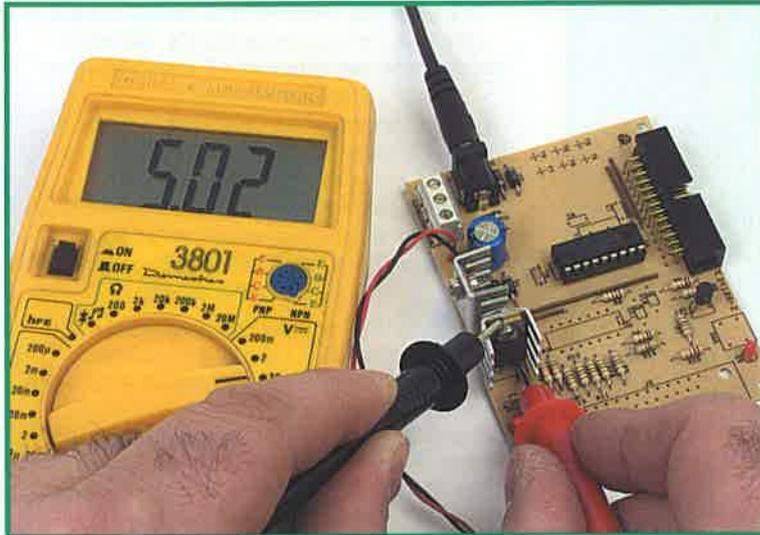
La stabilizzazione si realizza attraverso i due regolatori di tensione situati in U1 e U2. Lo stabilizzatore 7805 fornisce 5 V sull'uscita e con il 7812 otteniamo 12 V.

All'uscita dei 5 V di tensione di alimentazione del circuito, incontriamo il diodo led rosso D8. Questo diodo si illumina sempre quando la scheda è alimentata e se non ci sono cortocircuiti. Quindi se abbiamo la scheda alimentata, e questo diodo non si accende, dobbiamo spegnere immediatamente l'alimentazione, perché probabilmente esiste un problema.

Un buon riferimento di massa, per quando dobbiamo realizzare misure di tensione, è la parte metallica del regolatore 7805. Tutta questa parte si trova a potenziale 0 V.



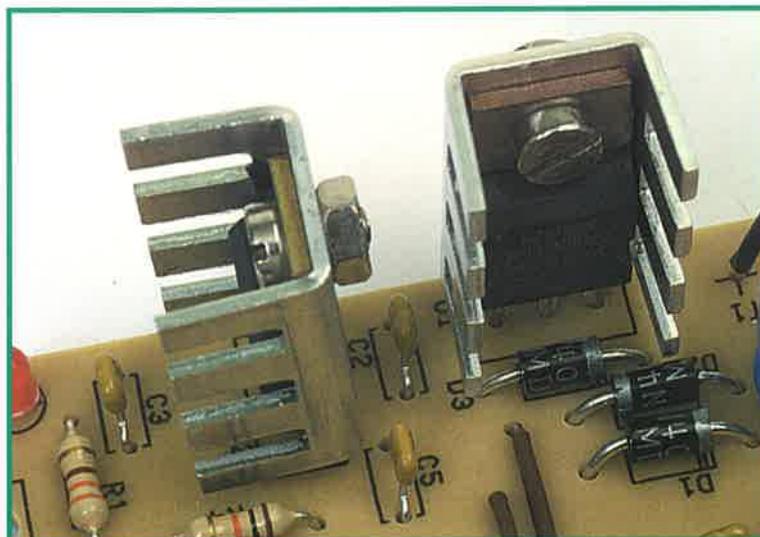
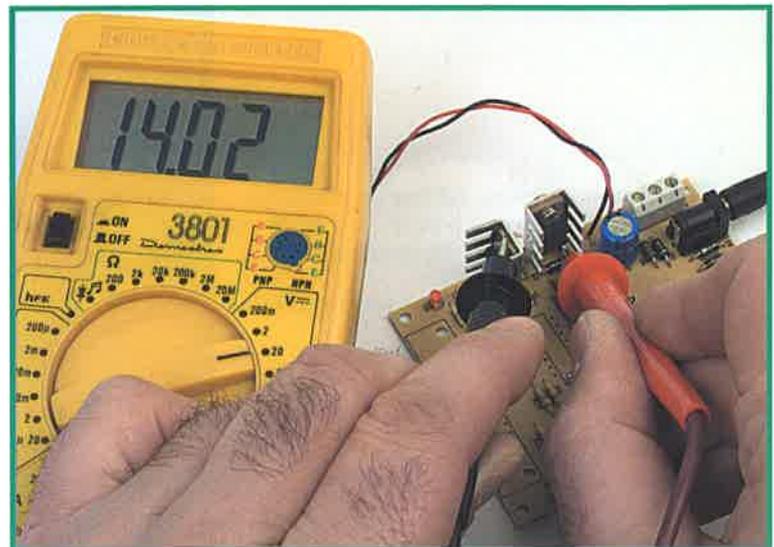
I regolatori 7805 e 7812 hanno la stessa piedinatura. Seguendo la numerazione mostrata nella figura il terminale 1 è sempre l'ingresso di tensione da stabilizzare. Questa tensione deve essere superiore ai 5 V nel caso del 7805 e superiore a 12 V nel caso del 7812. Il piedino centrale è a massa e sul piedino numero 3 otteniamo la tensione di uscita stabilizzata.



Quando abbiamo la scheda alimentata (tramite una pila o un trasformatore), possiamo verificare qual è la tensione di alimentazione che stiamo ottenendo. Per eseguire questa verifica posizioneremo il tester in misura di tensione sulla scala di 20 V, e punteremo i terminali come mostrato nell'immagine. Questa tensione sarà tanto migliore quanto si avvicinerà al valore ideale di 5 V.

Inoltre possiamo verificare qual è la tensione di uscita del regolatore 7812, posizionando i terminali sulla massa del circuito e sul piedino 3 del regolatore, come mostrato nella figura. Per questo è necessario alimentare la scheda con un trasformatore, o meglio con due pile da 9 V in serie.

Anche se la tensione dovrebbe essere di 12 V, vedremo che non sarà così, dato che sarà approssimativamente 13,8 V, la tensione necessaria per la programmazione. Questo si deve al fatto che sul piedino di massa di questo regolatore sono stati messi tre diodi rettificatori, D1-D2 e D3. I diodi hanno una caduta di tensione di circa 0,6 V, necessaria per la loro polarizzazione. Per questo, dato che i tre diodi sono messi in serie, la caduta totale è di 1,8 V, che sommata ai 12 V del regolatore rispetto a massa, ci fornisce i 13,8 V sull'uscita (piedino n°3).

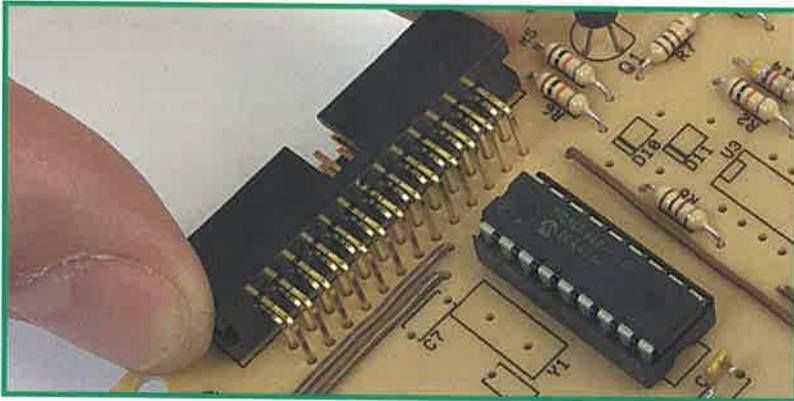


Notiamo anche che i regolatori hanno un'aletta di raffreddamento, perché quando assorbono molta corrente si surriscaldano. Possono fornire fino a 1 A. Questa perdita, sotto forma di calore è dovuta all'effetto Joule. La funzione dei condensatori situati sull'uscita dei regolatori, C2-C3, è quella di eliminare i possibili disturbi che possiamo avere sull'uscita dei regolatori, per tenere una tensione continua totalmente stabile e disaccoppiata.

MODULO DI CONTROLLO

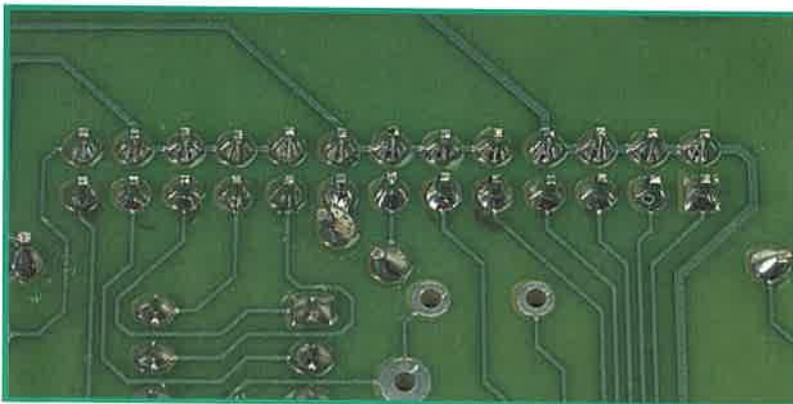
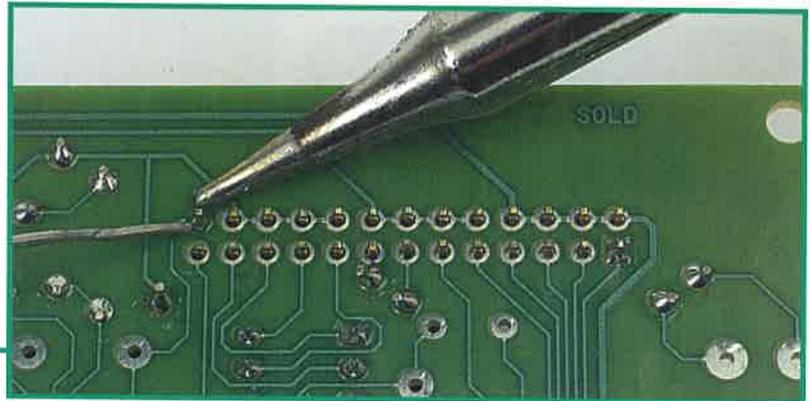
PL 14

Montaggio passo a passo



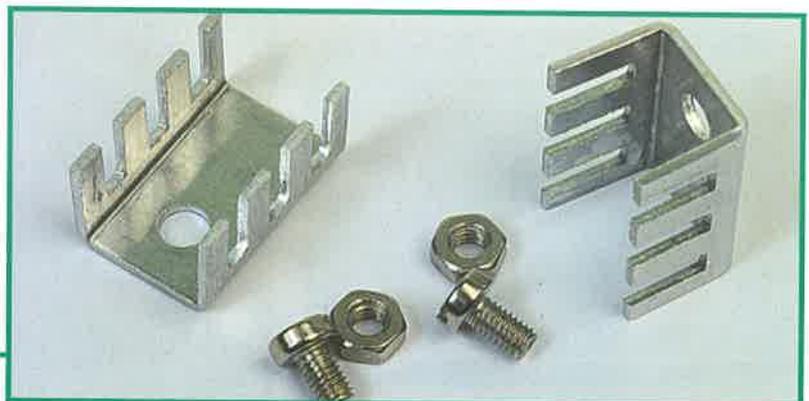
Per primo collegheremo il connettore a 26 pin ad angolo. Questo connettore farà sì che la scheda di controllo comunichi con il resto delle schede che formano il nostro robot. Dobbiamo introdurlo fino a quando non sarà a filo della scheda.

All'inizio salderemo solo i due angoli opposti del connettore, come avevamo già fatto con gli zoccoli. Poi verificheremo che il connettore sia ben inserito, prima di proseguire con il resto delle saldature.

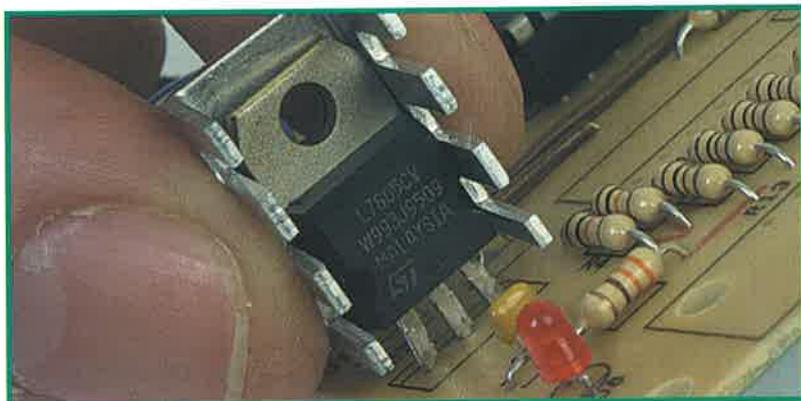


Dobbiamo saldare tutti i piedini, come mostrato nell'immagine, per ottenere delle saldature a forma di cono e che appoggino completamente sulla piazzola di saldatura. Dato che i piedini sono abbastanza vicini, dobbiamo fare attenzione a non provocare corto-circuiti.

I regolatori di tensione sono elementi che si riscaldano abbastanza, in funzione della quantità di corrente che assorbono. Per questo se vogliamo un funzionamento ottimale, dobbiamo montare delle alette di raffreddamento, perché dissipino bene il calore.

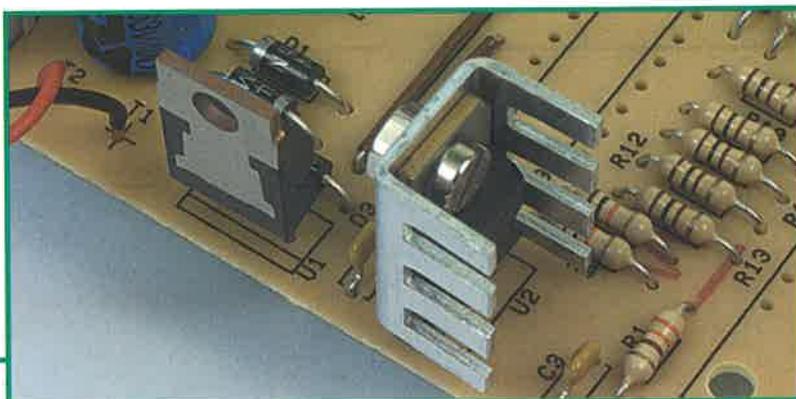


Montaggio passo a passo



I regolatori di tensione 7812 e 7805 sono situati verticalmente in U1 e U2, per cui utilizzeremo una vite per fissare le due alette di raffreddamento. Dobbiamo far coincidere il foro dell'aletta con quello del regolatore 7805, come si vede nell'immagine.

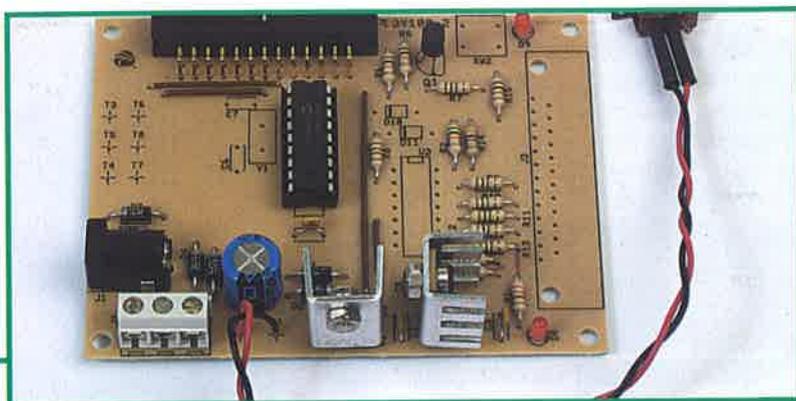
Adesso sistemeremo la vite e il bullone, per fissare le alette di raffreddamento al regolatore. Introduciamo la vite nella parte anteriore del regolatore, in modo da avvitare il bullone nella parte posteriore.



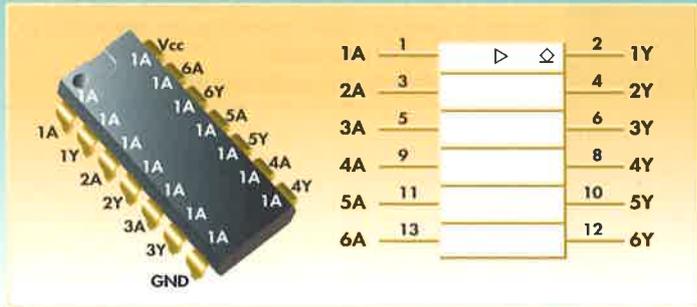
Lo stesso procedimento verrà ripetuto per montare le alette del regolatore 7812 inserito in U1. L'aletta deve essere orientata come mostra l'immagine.



Ecco l'aspetto finale che deve avere la scheda di controllo, una volta saldato il connettore e montate le alette di raffreddamento per i regolatori.

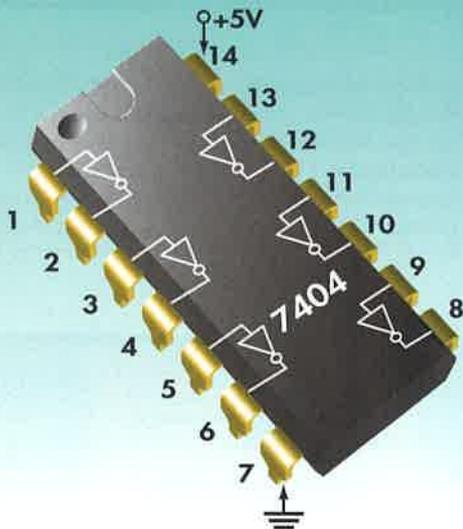
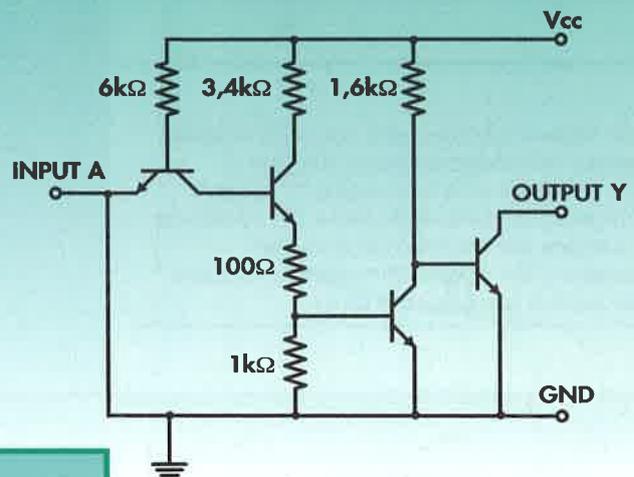


Consigli pratici



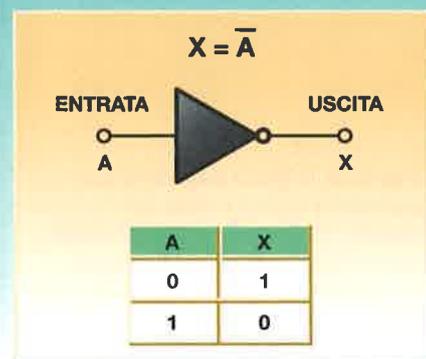
Il circuito integrato SN7407N, di cui è dotata la scheda di controllo, è un driver che utilizzeremo per amplificare livelli logici TTL. Ha un range di alimentazione che va da 4,75 V a 5,25 V. Dispone di sei ingressi e sei uscite secondo la configurazione illustrata nell'immagine.

Questo chip si utilizza per amplificare segnali TTL. Nella nostra scheda questa amplificazione si realizza tra i segnali che sono trasferiti dalla porta parallela del PC al microcontrollore. I segnali di uscita che si ottengono sono di 5 V o di 0 V (a seconda se l'ingresso sia un "1" logico o uno "0") con una capacità di corrente di circa 40 mA. Questa amplificazione è dovuta al fatto che possiede all'interno una configurazione basata su quattro transistor per uscita, come si vede nell'immagine. Le uscite sono a collettore aperto e richiedono quindi un collegamento tramite una resistenza di "pull-up" al positivo.



Altri circuiti logici: i circuiti logici sono molto utilizzati in elettronica. Permettono di realizzare funzioni logiche tra segnali elettrici. Quando sapremo programmare il microcontroller, vedremo che potremo realizzare tramite software molte operazioni logiche, che in altro modo richiederebbero l'impiego di numerosi circuiti logici. Il circuito dell'immagine contiene sei funzioni logiche di tipo NOT.

Come si vede nell'immagine, ognuna delle uscite rappresenta lo stato inverso ai corrispondenti ingressi. Questo circuito integrato nega all'uscita il livello logico che vede all'ingresso. La tabella della verità esposta, riassume il funzionamento di ognuna delle funzioni NOT.



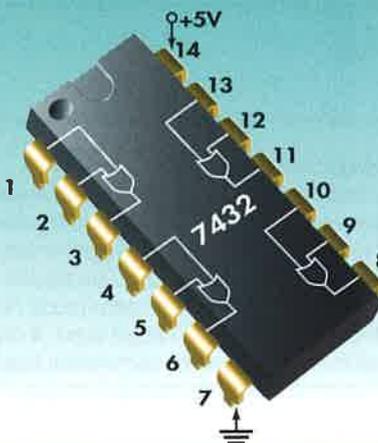
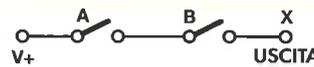


La funzione logica AND. L'immagine mostra il contenitore del circuito integrato SN7408. L'uscita va a livello logico 1 solo se c'è un 1 in entrambi i suoi ingressi. Essa si comporta come se i suoi ingressi fossero interruttori in serie. Entrambi gli interruttori devono essere attivi nel medesimo tempo.

La figura mostra il simbolo logico, così come la tabella della verità a cui corrisponde questa funzione. È mostrata l'uscita che va a livello logico "1", quando entrambi gli ingressi sono anche loro a "1". Lo schema elettrico mostra due interruttori in serie, che rappresentano i due ingressi. Entrambi devono essere attivi per avere la tensione +V in uscita.



A	B	X
0	0	0
1	0	0
0	1	0
1	1	1

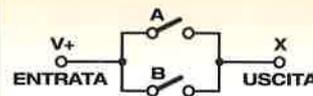


La funzione logica OR. La figura mostra il contenitore del circuito integrato SN7432. Questo integrato contiene quattro funzioni logiche OR. L'uscita di ognuna di queste va a livello logico "1" quando uno qualsiasi dei suoi ingressi vale "1". Il suo comportamento equivale elettricamente a due interruttori collegati in parallelo.

La figura mostra il simbolo, la tavola della verità e lo schema elettrico equivalente della funzione OR. Guardando sia la tabella che il circuito possiamo notare che quando si attiva uno qualunque degli ingressi, si attiva anche l'uscita. Queste tre funzioni logiche base che abbiamo visto, si possono combinare tra di loro per realizzare altri tipi di funzioni: NAND, NOR, XOR, e XNOR.



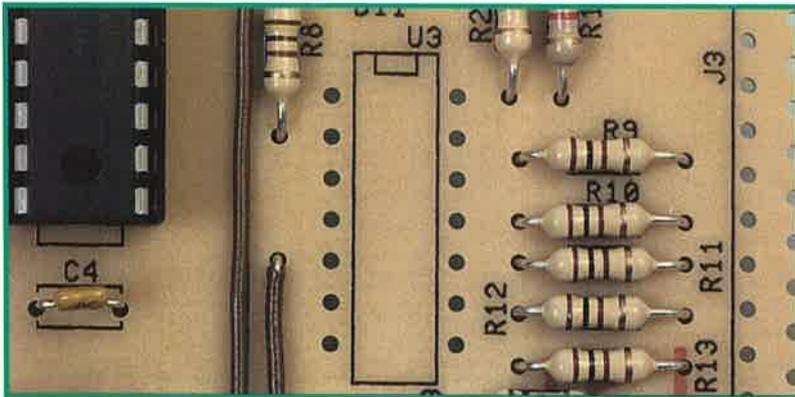
A	B	X
0	0	0
1	0	1
0	1	1
1	1	1



MODULO DI CONTROLLO

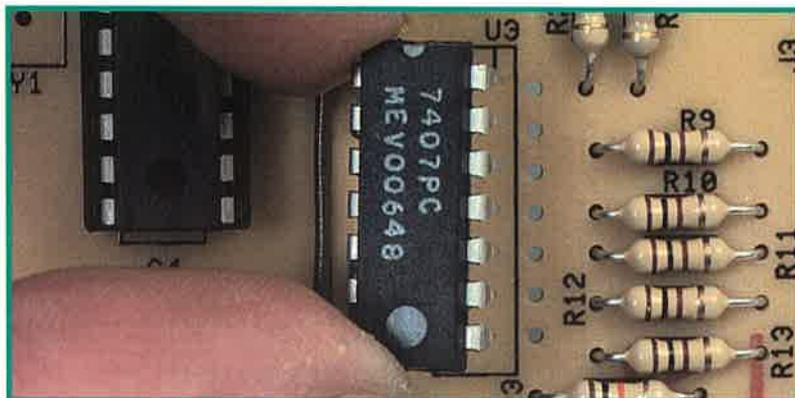
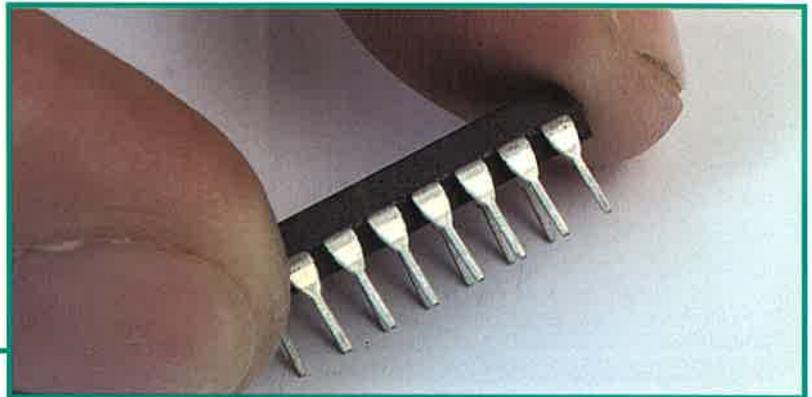
PL 16

Montaggio passo a passo



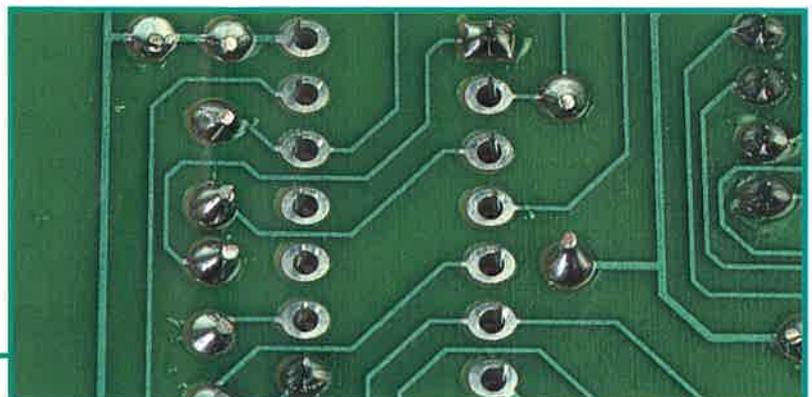
Monteremo in primo luogo il circuito integrato SN7407, il quale deve essere collocato in U3. Il verso di inserzione del chip è segnalato con un quadratino nella parte superiore del rettangolo disegnato nella serigrafia.

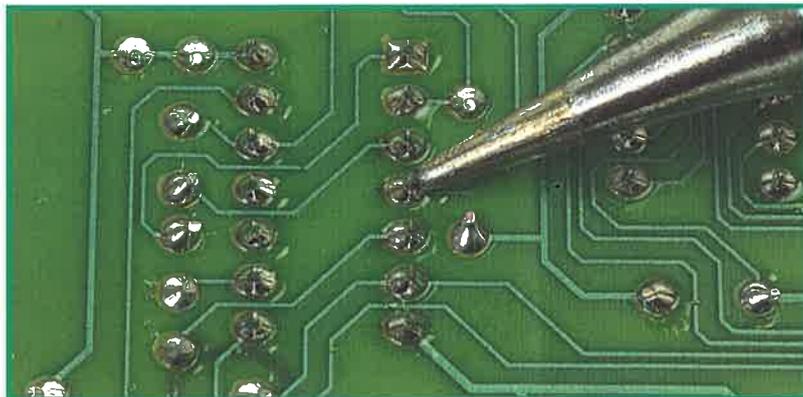
Per inserire il chip nella scheda dovremo piegare un poco i piedini. Per questo raccomandiamo di fare una leggera pressione, sopra la superficie liscia, come si vede nell'immagine.



Una volta piegati leggermente i piedini inseriremo il chip nella scheda, badando di far coincidere la tacca di riferimento che si trova su uno degli estremi del chip, con il quadratino disegnato nella serigrafia.

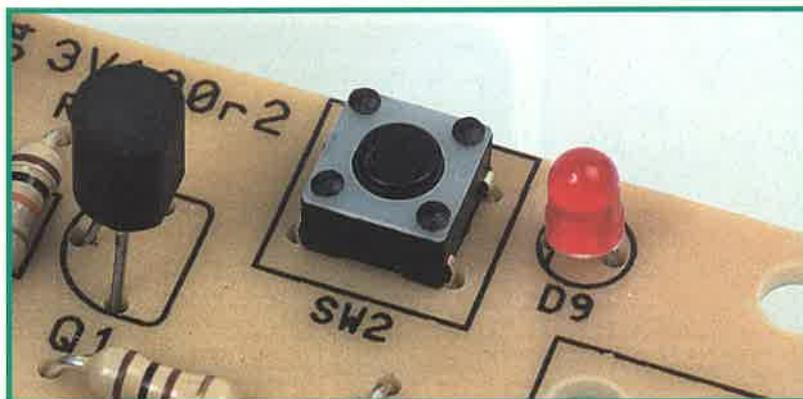
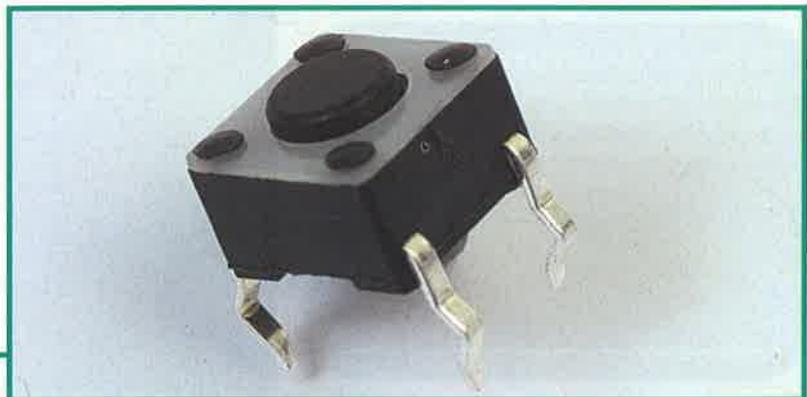
All'inizio salderemo solo i due terminali opposti del chip; prima di proseguire con le altre saldature dobbiamo verificare se il chip è a filo della scheda e con il verso corretto.





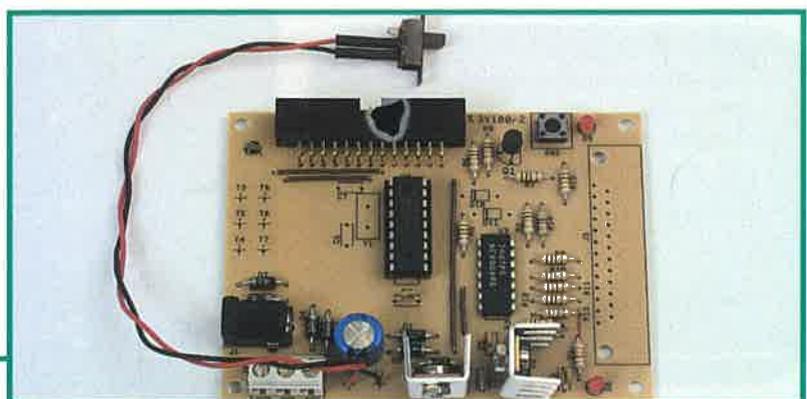
Quando saremo sicuri che l'inserzione del chip sia corretta, salderemo i piedini restanti. Ricordiamo che le saldature non devono avere la forma di bolla, ma devono avere una forma di cono appoggiato sulla superficie della scheda. Inoltre non dobbiamo porre una quantità eccessiva di stagno e controllare che l'eccesso di calore non rovini il circuito. Conviene precedere con attenzione.

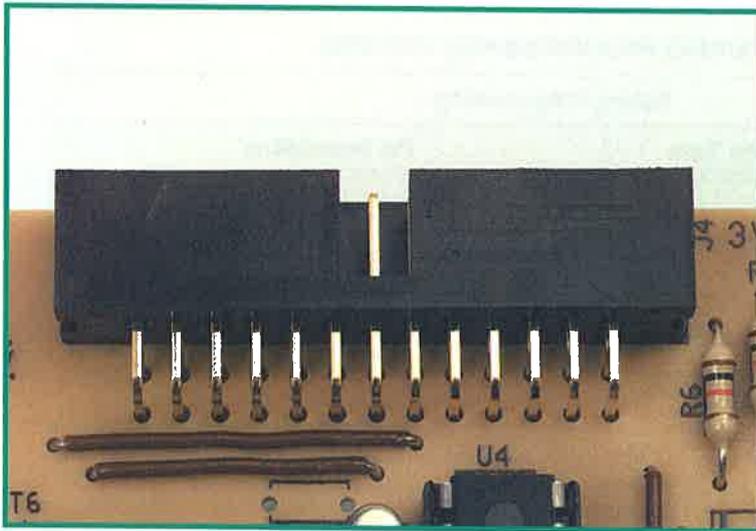
Ora salderemo il pulsante miniaturizzato a circuito normalmente aperto. Per inserirlo nella scheda è necessario prima raddrizzare i suoi piedini. Per questo raccomandiamo l'utilizzo di una pinzetta a becco piatto.



Dobbiamo inserire il pulsante in SW2 con il verso mostrato nell'immagine, altrimenti la scheda di controllo non funzionerà. Come si può vedere il pulsante non ha tutti e quattro i lati uguali. Due possiedono piedini di uscita, gli altri due no.

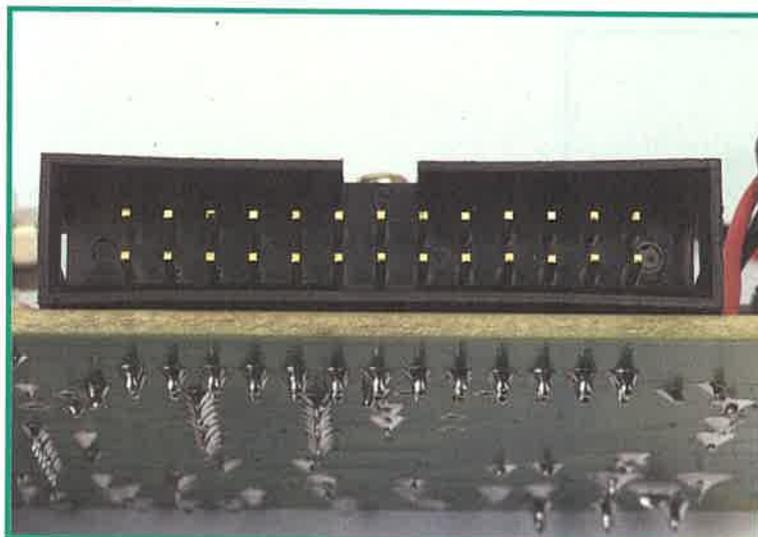
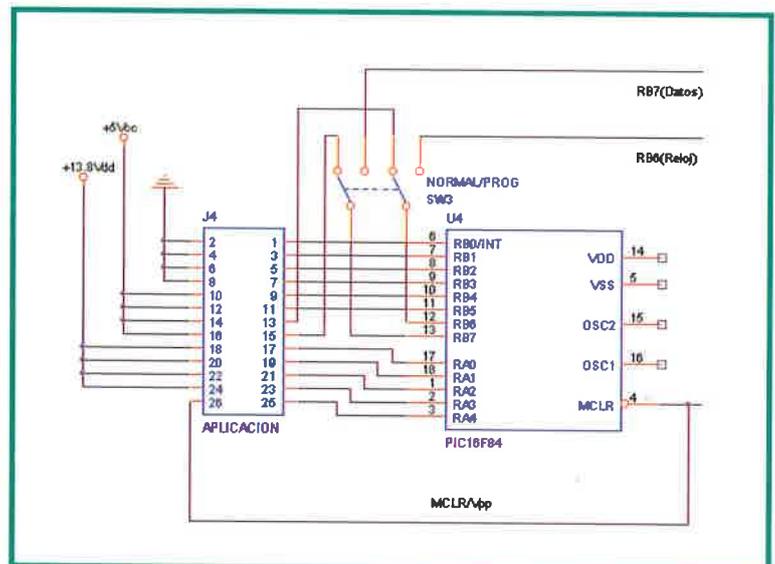
Dopo aver ultimato il montaggio del circuito integrato U3 e del pulsante SW2, dobbiamo assicurarci che la scheda abbia montati tutti i componenti illustrati nella figura, e con il corretto orientamento.





Il connettore d'applicazione J4 lo chiameremo PIC-BUS. Contiene tutti i segnali d'ingresso e uscita del microcontroller PIC situato nella scheda di controllo. Attraverso esso ci collegheremo con il resto delle schede da cui è costituito Monty. Il PIC potrà operare con sensori e periferiche situati in queste schede senza la necessità di muoverlo dal suo zoccolo.

Come si può osservare nello schema, attraverso questo connettore passano anche i segnali di alimentazione da 5 V e 13 V, e la massa. Fondamentale in un progetto elettronico è che tutte le masse siano comuni.



Il connettore ha 26 piedini. Nella fila superiore ci sono quelli dispari dal 1 al 25 iniziando da destra. Nella fila inferiore ci sono quelli pari dal 2 al 26 iniziando sempre da destra. Per esempio, i segnali della porta B sono situati nella fila superiore del connettore, in ordine, dal piedino 1 al 15 iniziando da destra. Il nostro microcontroller PIC16F84, permette la memorizzazione seriale ICSP (In Circuit Serial Programming), grazie alla quale servono solo due pin per introdurre il programma (RB6 e RB7).

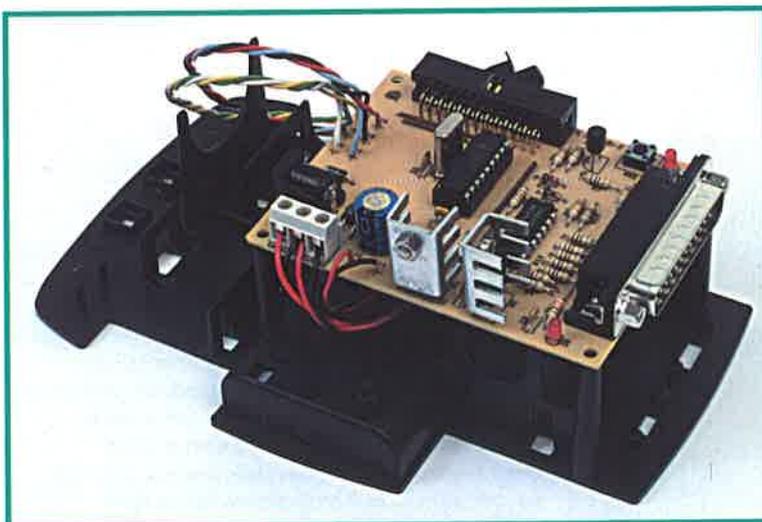
PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F8X

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
$\overline{\text{MCLR}}$	VTEST MODE	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = Input, O = Output, P = Power

Come mostrato nella tabella, la programmazione si realizza con i pin RB6 e RB7, il primo è quello che trasferisce il segnale di clock, e RB7 porta i bits da leggere e scrivere. Questo è il motivo per cui serve il commutatore a due posizioni SW3. Durante la programmazione dobbiamo scollegare i pin RB6 e RB7 dal PIC-BUS. Per questo apriremo il circuito con il commutatore SW3 (Posizione PROG). Una volta memorizzato chiuderemo il commutatore per collegare le linee RB6 e RB7 di nuovo al PIC-BUS, potendo in questo modo utilizzarle come ingressi e uscite digitali (Posizione RUN).

In seguito la scheda di controllo verrà montata sul telaio di Monty. Il commutatore SW3 verrà alloggiato nella scanalatura corrispondente come si vede nella figura. Quando SW3 sarà nella posizione PROG, il PIC rimarrà in modo programmazione. I segnali RB6 e RB7 saranno scollegati dal PIC-BUS. Una volta memorizzato SW3 si passa in posizione RUN.

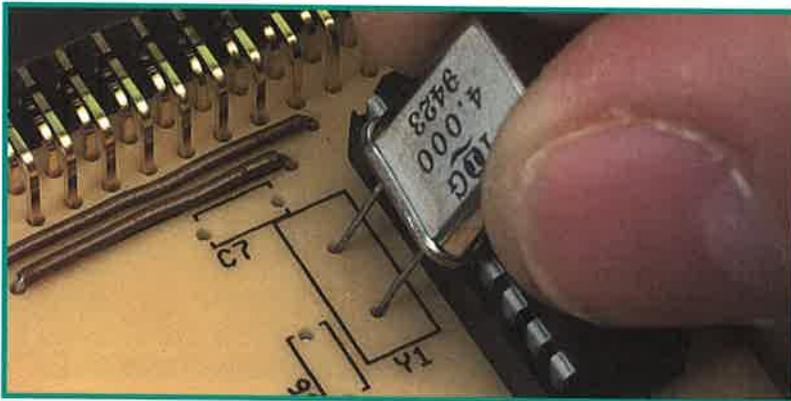


La figura mostra la disposizione finale della scheda di controllo sopra il telaio di Monty. Anche se nel passo a passo si spiega come fissare SW3 su questo telaio, si raccomanda di non farlo in questo momento. Potremo continuare il montaggio più comodamente.

MODULO DI CONTROLLO

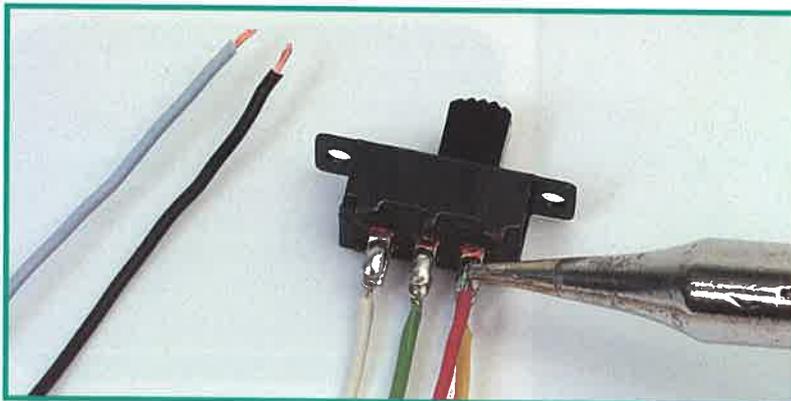
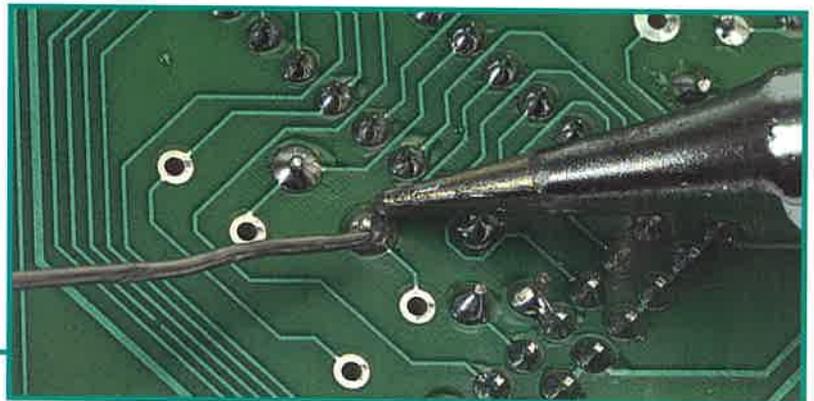
PL 18

Montaggio passo a passo



In primo luogo salderemo il cristallo di quarzo. Questo componente deve essere inserito in Y1. Il cristallo di quarzo non ha polarità, per cui è indifferente il verso in cui lo si introduce.

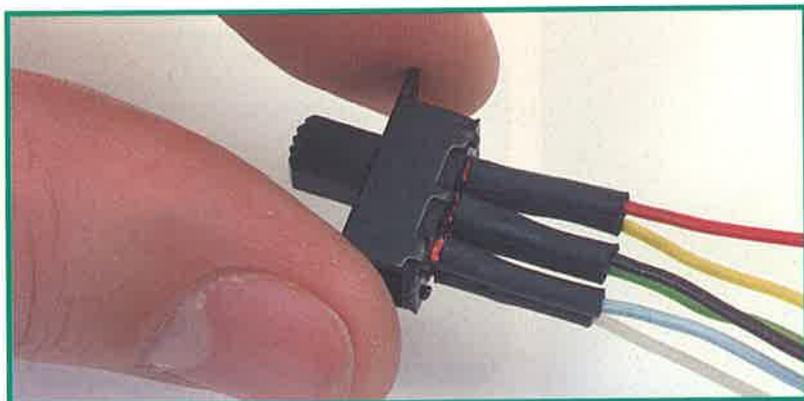
Lo spazio di saldatura che dobbiamo riempire per saldare il cristallo di quarzo, è un po' più grande di quello degli altri componenti che abbiamo montato fino ad ora, come resistenze o zoccoli. Dobbiamo riempire con lo stagno tutta la saldatura.



Ora salderemo i cavi del commutatore a slitta a due posizioni. Speleremo i cavi da ambo gli estremi con l'aiuto delle forbici o di uno spelafili.

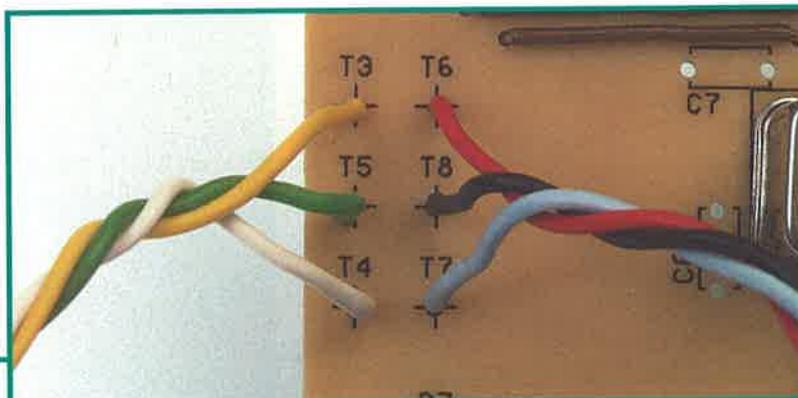
Quando avremo pelato i sei estremi dei cavi, li salderemo ai sei terminali dell'interruttore, e monteremo poi il termorestringente, il quale eviterà che i piedini dell'interruttore entrino in contatto fra loro. Il termorestringente deve essere montato e scaldato con la punta del saldatore fino a quando aderirà totalmente al piedino.





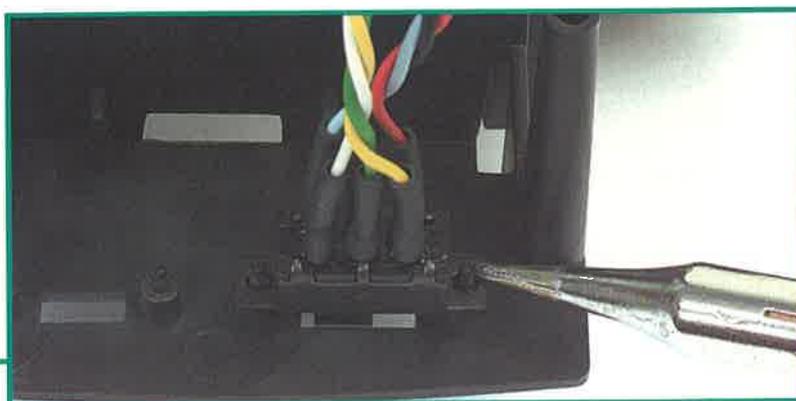
Ecco come deve rimanere l'interruttore, una volta montati tutti i termorestringenti. Forse dovremo separare leggermente i due fili dell'interruttore per poter introdurre il termorestringente e saldarlo bene.

Ora salderemo l'interruttore alla scheda di controllo. Prima attorciglieremo i cavi di ogni circuito di commutazione. Introduciamo i cavi dalla parte superiore della scheda, in modo ordinato, come si vede nell'immagine, e realizzeremo la saldatura dall'altra parte.



Questo interruttore è collocato sul telaio del robot. Dovremo montarlo nella fessura più grande, situata nella parte anteriore del medesimo. Per fare in modo che l'azionamento dell'interruttore corrisponda con le parole PROG/RUN segnate sul telaio, dovremo introdurlo così come mostrato nell'immagine.

Se vogliamo fissare l'interruttore al telaio, dobbiamo introdurlo completamente, facendo coincidere gli agganci del telaio con i due fori dell'interruttore. Una volta introdotto riscaldiamo la plastica con il saldatore per fissare la posizione dell'interruttore.



Il microcontroller PIC16F84 può funzionare con quattro diversi tipi di oscillatori, che possono essere specificati dall'utilizzatore al momento della programmazione, mediante la parola di configurazione. Nella nostra scheda di controllo abbiamo utilizzato un oscillatore tipo XT, formato da un cristallo di quarzo e da due condensatori ceramici da 27 pF.

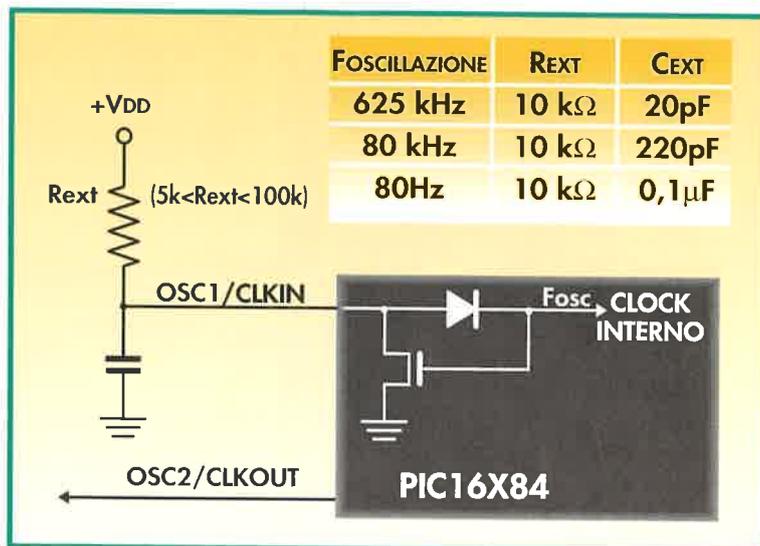
I quattro tipi di oscillatori permessi sono:

1°) Oscillatore tipo RC

Questo tipo di oscillatore, di costo molto basso, è formato semplicemente da una resistenza e da un condensatore. Il collegamento di questo oscillatore è riportato nel disegno in alto, inoltre sono anche indicate alcune coppie di valori per ottenere determinate frequenze di oscillazione.

2°) Oscillatore tipo XT

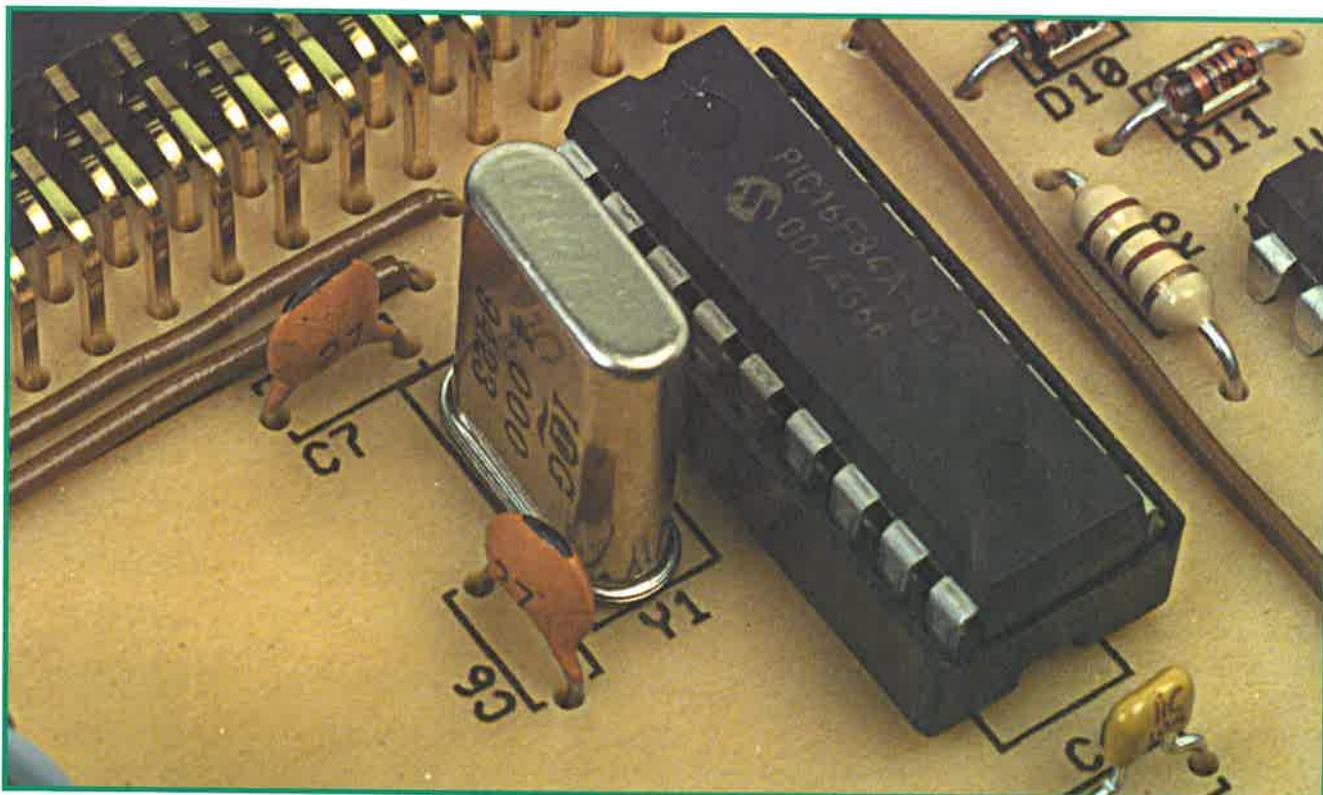
Questo è l'oscillatore utilizzato nella nostra scheda, e si basa su di un oscillatore di cristallo, o risonatore, per



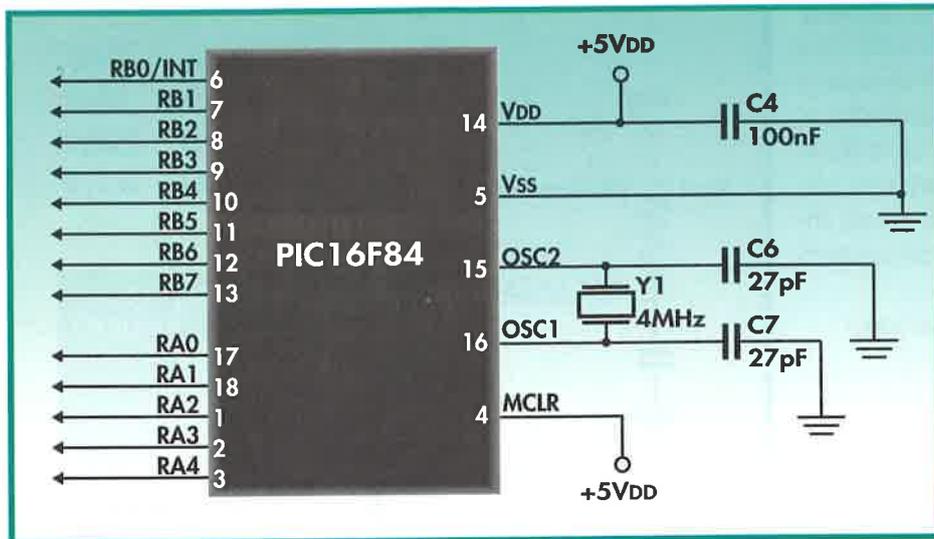
Collegamento e valori tipici per un oscillatore RC di basso costo.

frequenze comprese fra 100 KHz e 4 MHz. Nella figura riportata in alto nella pagina seguente possiamo vedere lo schema elettrico per il collegamento di questo oscillatore al nostro microcontroller sulla scheda di controllo del Monty.

Con questo oscillatore otteniamo una frequenza di



Oscillatore tipo XT utilizzato nella nostra scheda.



Schema elettrico per il collegamento di un oscillatore tipo XT. È quello impiegato nella scheda di controllo del Monty.

4 MHz con la quale il PIC esegue un'istruzione di programma ogni microsecondo.

3°) Oscillatore tipo HS

Si tratta di un oscillatore che raggiunge un'alta velocità, compresa fra 4 e 20 MHz, basato su di un cristallo di quarzo o un risonatore ceramico. È il tipo di oscillatore da utilizzare per raggiungere le maggiori velocità di esecuzione. Si collega in parallelo con due condensatori, come il tipo XT. 20 MHz è anche la massima frequenza a cui può lavorare il PIC, che a questa velocità esegue un'istruzione in 200 ns.

4°) Oscillatore tipo LP

Oscillatore di basso consumo, che utilizza un quarzo e che può lavorare nel range di frequenze da 35 a 200 KHz. Si collega allo stesso modo dei tipo XT o HS. Nella tabella sono riportati i valori dei condensatori che dobbiamo utilizzare per ottenere le frequenze desiderate, nel caso di oscillatori tipo XT, HS o LP, con cristalli di quarzo. La frequenza di funzionamento del microcontroller è un parametro fondamentale, da cui dipende la velocità di esecuzione

del programma, e anche il consumo di energia. All'interno della stessa famiglia dei PIC esistono modelli progettati per lavorare a frequenze differenti. Il tempo di esecuzione di un'istruzione è quattro volte il periodo di clock dell'oscillatore che stiamo utilizzando.

$$\text{TCICLO DI ISTRUZIONE} = 4 * \text{TCLOCK}$$

Le istruzioni di salto, che ci serviranno per testare lo stato dei bit, impiegano il doppio del tempo ad essere eseguite rispetto ad un'istruzione semplice.

Per esempio, se abbiamo un oscillatore da 4 MHz, un programma da 500 istruzioni, 10 delle quali sono di salto, il programma impiegherà ad essere eseguito:

$$\text{TCICLO DI ISTRUZIONE} = 4 * \text{TCLOCK} = 4 * 1/(4 * 10^6) = 1 \mu\text{s}$$

$$T_{\text{programma}} = 490 * 1 \mu\text{s} + 10 * 2 \mu\text{s} = 510 \mu\text{s}$$

Come prima valutazione di questo risultato possiamo osservare che i programmi che introduciamo nel microcontrollore vengono eseguiti ad altissima velocità, praticamente in tempo reale.

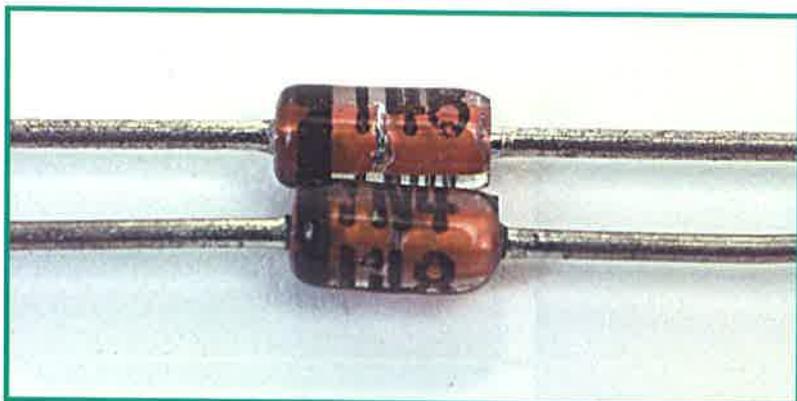
MODE	FREQ.	PIC16CR84	PIC16F83
LP	32kHz	68-100 pF	68-100 pF
	200 kHz	15-33 pF	15-33 pF
XT	100 kHz	100-150 pF	100-150 pF
	2 MHz	15-33 pF	15-33 pF
	4 MHz	15-33 pF	15-33 pF
HS	4 MHz	15-33 pF	15-33 pF
	10 MHz	15-33 pF	15-33 pF

Tabella riassuntiva con i valori dei condensatori proposti da Microchip per differenti frequenze di lavoro e tipi di oscillatori.

MODULO DI CONTROLLO

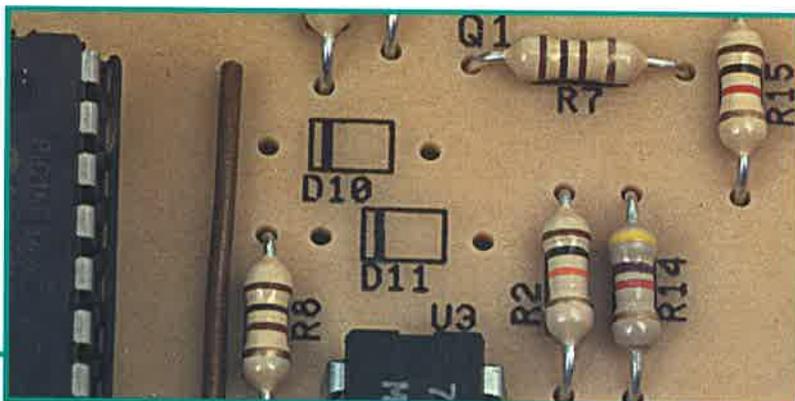
PL 20

Montaggio passo a passo



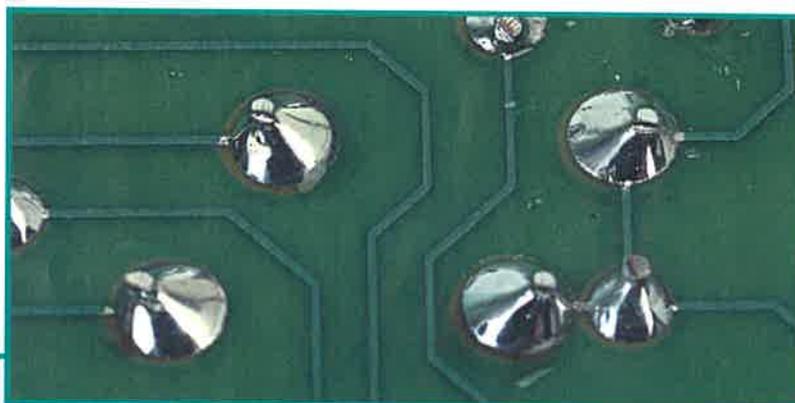
Dobbiamo nuovamente saldare dei diodi: in questo caso si tratta di diodi di segnale, anche loro dotati di polarità. Questi diodi sono di colore arancio e la polarità viene indicata con una banda nera su di un estremo, che indica il catodo o terminale negativo del diodo.

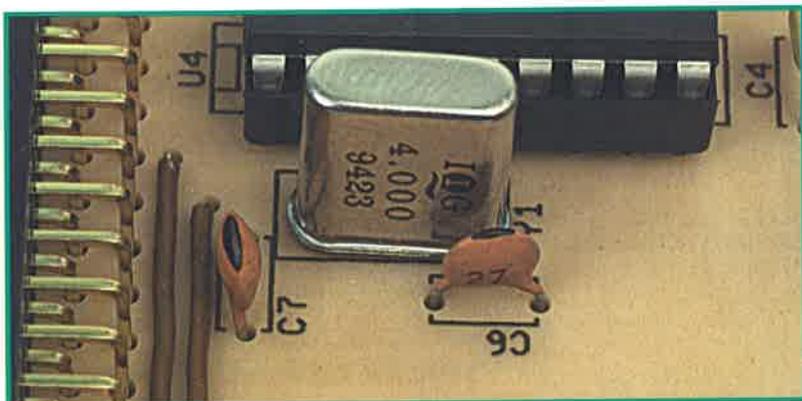
I due diodi devono essere collocati in D10 e D11, nella scheda inoltre è segnalata anche la loro polarità, con una doppia linea nera disegnata dalla parte dove dovremo montare il catodo dei due diodi.



Questa è la posizione corretta per l'inserzione dei diodi. Devono essere a filo della scheda con il catodo che coincide con la doppia linea nera della serigrafia.

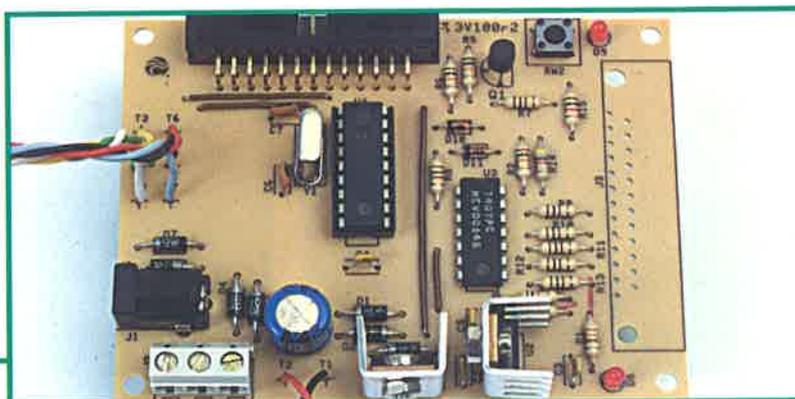
Le saldature dei due diodi sono un poco più grandi di quelle delle resistenze, dobbiamo riempire con lo stagno tutta la piazzola di saldatura formando un cono.





I componenti successivi da montare sono i condensatori da 27 pF. Dovranno essere inseriti in C6 e C7, a lato del cristallo di quarzo. Questi condensatori non hanno polarità, per cui è indifferente il verso in cui vengono montati.

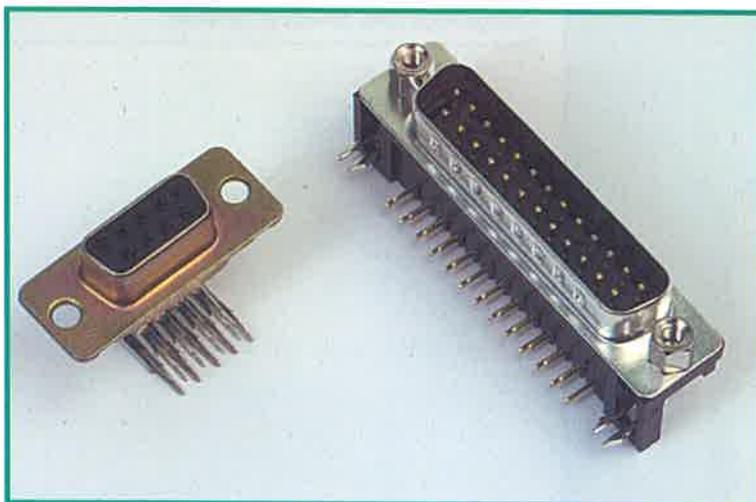
Dopo aver saldato i diodi e i condensatori, la scheda di controllo avrà questo aspetto. Ci rimane solamente da saldare un connettore per comunicare con il PC, quindi avremo terminato il montaggio.



Il connettore della pila è per i modelli da 9 V. Il telaio di Monty possiede, nella sua parte inferiore, un incavo in cui potremo alloggiare questa pila; inoltre possiede un'apertura dalla quale potremo far uscire i cavi all'esterno, anche con il coperchio chiuso.

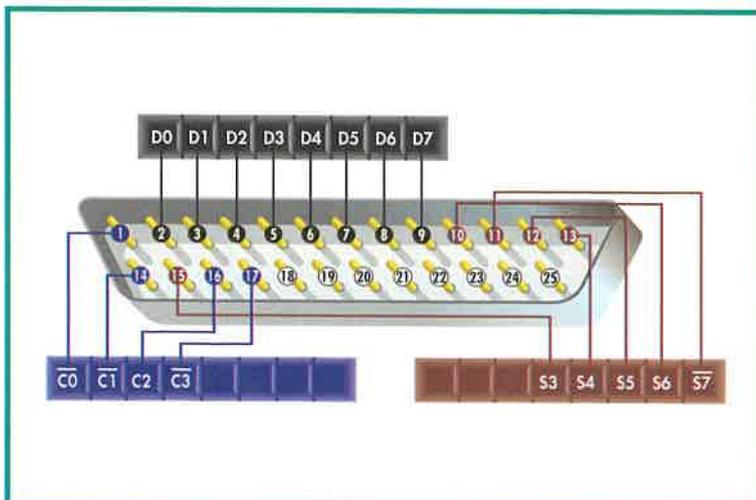
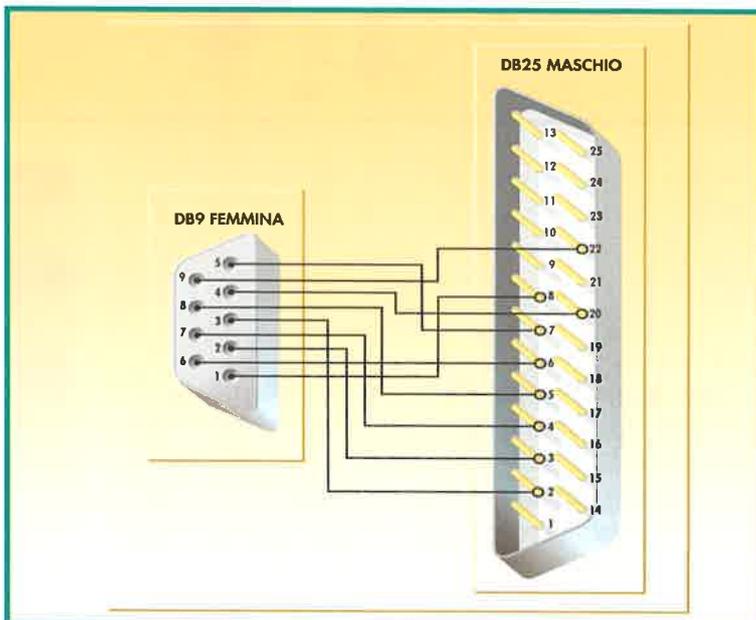
Questa ruota è una delle quattro che comporranno la base di movimento del robot. Due di queste le collegheremo direttamente ai motori, le altre serviranno da appoggio, e saranno unite tramite un asse.



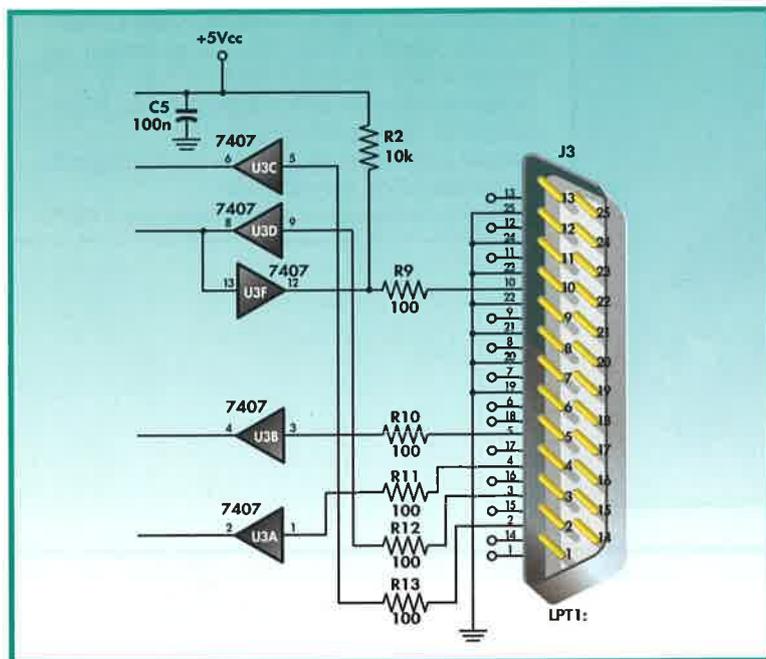


Connettori DB9 e DB25. I connettori DB9 e DB25 sono uno standard molto diffuso in elettronica, per vari motivi, uno dei quali è che sono il tipo di connettore utilizzato dalla maggioranza dei PC per comunicare con le periferiche tipo il mouse o la stampante. Le porte seriali utilizzano il connettore DB9, mentre quelle parallele (e anche alcune seriali) impiegano il DB25.

Corrispondenze DB9 – DB25. Quando si utilizza il connettore DB25 per trasmissioni seriali, si impiega una corrispondenza standard tra i due connettori DB9 e DB25, che è quella mostrata nell'immagine. Porta parallela. È il connettore DB25 montato nella nostra scheda di controllo e comunicherà con la porta parallela del PC. Questa porta usa livelli di tensione TTL, quindi non sarà necessario adattare i suoi livelli alla tensione della scheda di controllo. Comunque, è stato utilizzato un circuito 7407 per amplificare i segnali. Usando il connettore DB25 come porta parallela, la configurazione dei suoi pins è la seguente:



Pins del connettore DB25 (porta parallela). Le connessioni illustrate nella figura sono quelle realizzate nella scheda di controllo sul connettore DB25 maschio e il chip 7407 (che comunicherà con il microcontrollore).



Collegamenti nella scheda di controllo.

Tutte le connessioni che si vedono nella tabella si impiegano quando la porta si utilizza per collegarsi, per esempio, ad una stampante. Abbiamo già visto che il nostro microcontrollore si scrive mediante ICSP, per cui utilizzeremo solo pochi pins per la comunicazione fra il PC e il microcontrollore.

Se osserviamo lo schema elettronico della figura, vedremo che tutti i segnali che vanno verso il connettore, hanno una resistenza seriale da 100 Ω, che serve per la protezione della porta contro le sovratensioni. Inoltre possiamo vedere che l'unico ingresso utilizzato verso il PC è il piedino 10, attraverso il quale si potrà leggere il contenuto del PIC.

I piedini 5, 4, 3 e 2, anch'essi usati, sono uscite in arrivo dal PC che realizzano distinte funzioni durante la memorizzazione. Per esempio, il piedino 2 porterà il segnale periodico del clock, e il piedino 3 i bits da scrivere nel PIC. Bisogna inoltre dire che i segnali di terra del connettore 18-25, sono comuni con la terra della scheda di controllo.

I segnali della porta parallela che corrispondono con la disposizione dei pins del connettore, sono indicati nella tabella a lato.

Nome del	Bit	n° pin	Direzione
Segnale			I/O
-Strobe	-C0	1	Output
+Data Bit 0	D0	2	Output
+Data Bit 1	D1	3	Output
+Data Bit 2	D2	4	Output
+Data Bit 3	D3	5	Output
+Data Bit 4	D4	6	Output
+Data Bit 5	D5	7	Output
+Data Bit 6	D6	8	Output
+Data Bit 7	D7	9	Output
-Acknowledge	S6	10	Input
+Busy	-S7	11	Input
+Paper End	S5	12	Input
+Select In	S4	13	Input
-Auto Feed	-C1	14	Output
-Error	S3	15	Input
-Initialize	C2	16	Output
-Select	C3	17	Output
Ground		-18-25-	

MODULO DI CONTROLLO

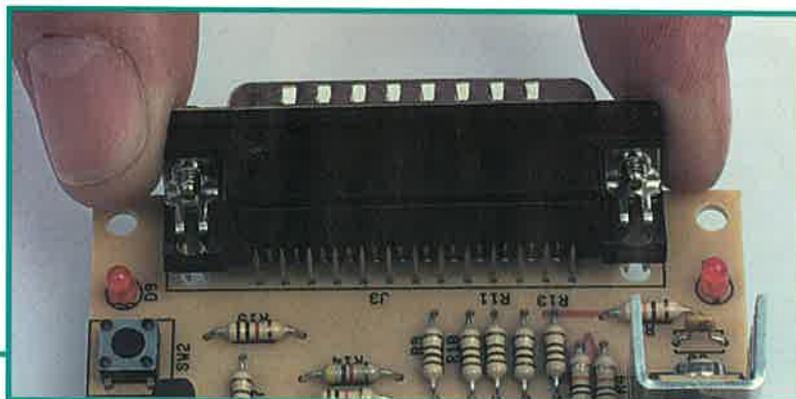
PL 22

Montaggio passo a passo

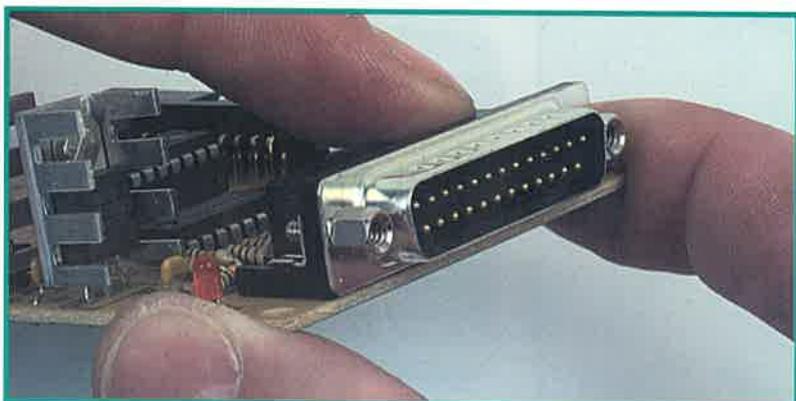


Il connettore DB25 che dobbiamo saldare, è un tipo di connettore standard di comunicazione. Tutti i PC ne hanno uno che di solito è utilizzato per collegare la stampante o lo scanner.

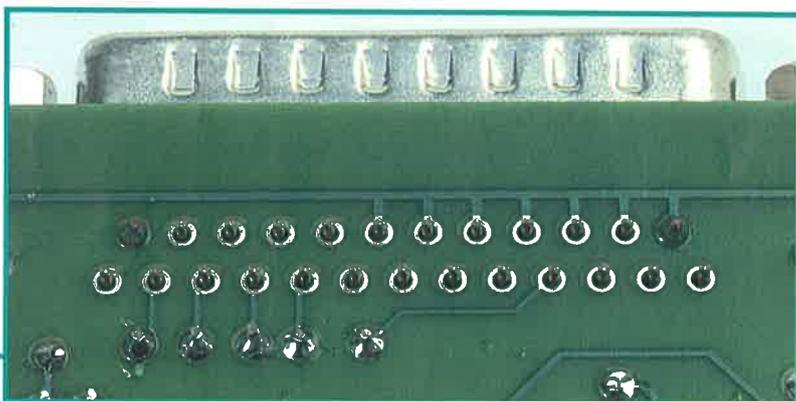
Dobbiamo introdurre il connettore facendo coincidere non solo i pins, ma anche i due supporti che ci sono ai suoi lati, con i fori presenti sulla scheda. Questi supporti assicurano una buona stabilità del connettore in modo che le saldature non si danneggino quando lo solleciteremo per collegare o scollegare dei cavi.

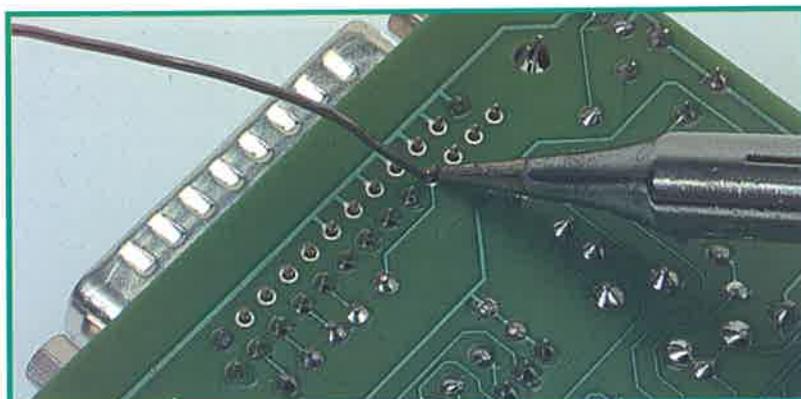


Dobbiamo premere il connettore fino a che sarà completamente incastrato nella scheda, come si vede nell'immagine.



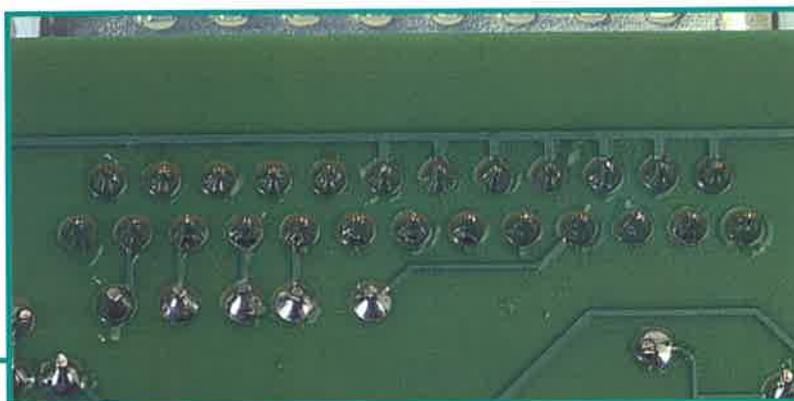
Quando avremo posizionato bene il sensore, procederemo a saldarlo, iniziando dai due pins agli angoli. Dopo aver realizzato queste due prime saldature, verificheremo che il connettore non si sia mosso, e sia ben inserito.



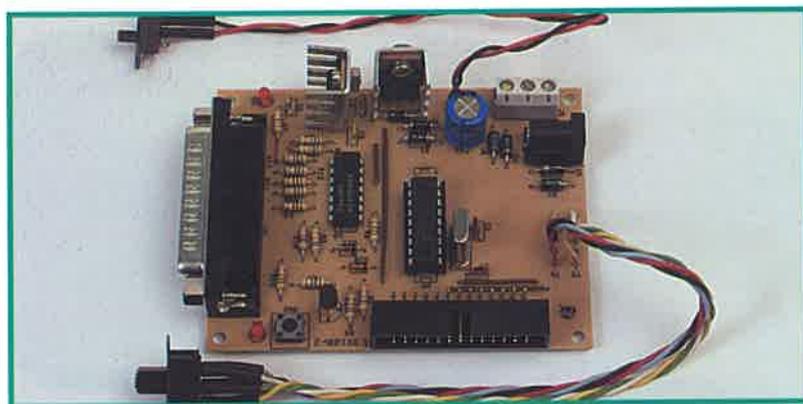


Dopo salderemo il resto dei pins del connettore. Raccomandiamo di saldare prima una linea di pins completa, per esempio quella inferiore, e poi l'altra linea. Le saldature non devono avere troppo stagno, per evitare che uno o più piedini entrino in contatto fra di loro.

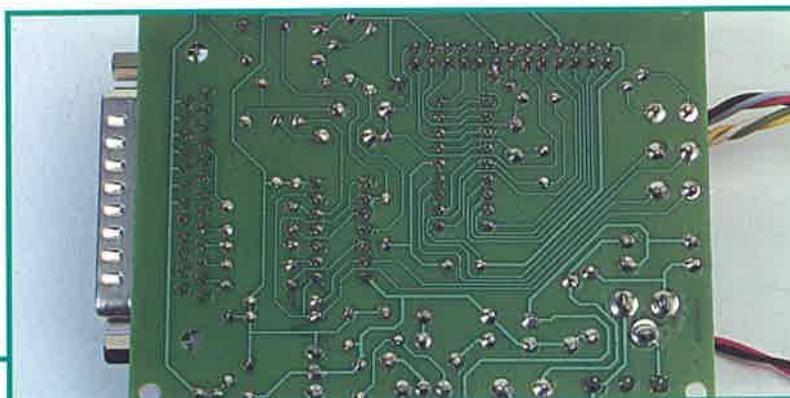
Questo è l'aspetto che devono avere tutte le saldature del connettore. Devono formare piccoli coni appoggiati alla superficie, e dovremo assicurarci che non entrino in contatto le saldature dei piedini.

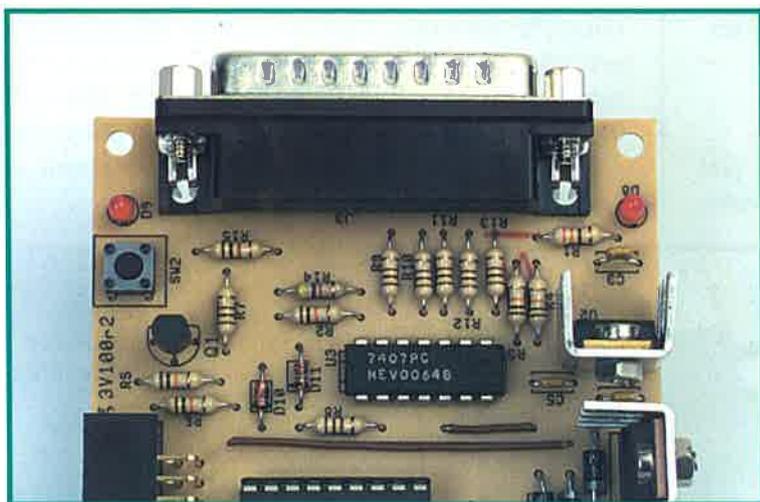


Abbiamo così terminato il montaggio della scheda di controllo. Questa scheda ci servirà per programmare il microcontrollore, e comunicherà attraverso il PIC-BUS con il resto delle schede che formeranno l'elettronica di Monty.



Al termine, le saldature della scheda devono presentare un aspetto simile a quello dell'immagine. Devono essere brillanti, non avere troppo stagno e formare coni appoggiati sulla scheda. Potremo utilizzare la funzione di continuità del tester per verificare questo fra le saldature.





Nell'immagine è mostrato lo schema completo che si utilizza per la programmazione del microcontroller PIC. Attraverso il connettore DB25 (J3) ci collegheremo alla porta parallela del PC, e scaricheremo i programmi nel microcontroller. Inoltre potremo leggere i programmi che il PIC ha in memoria.

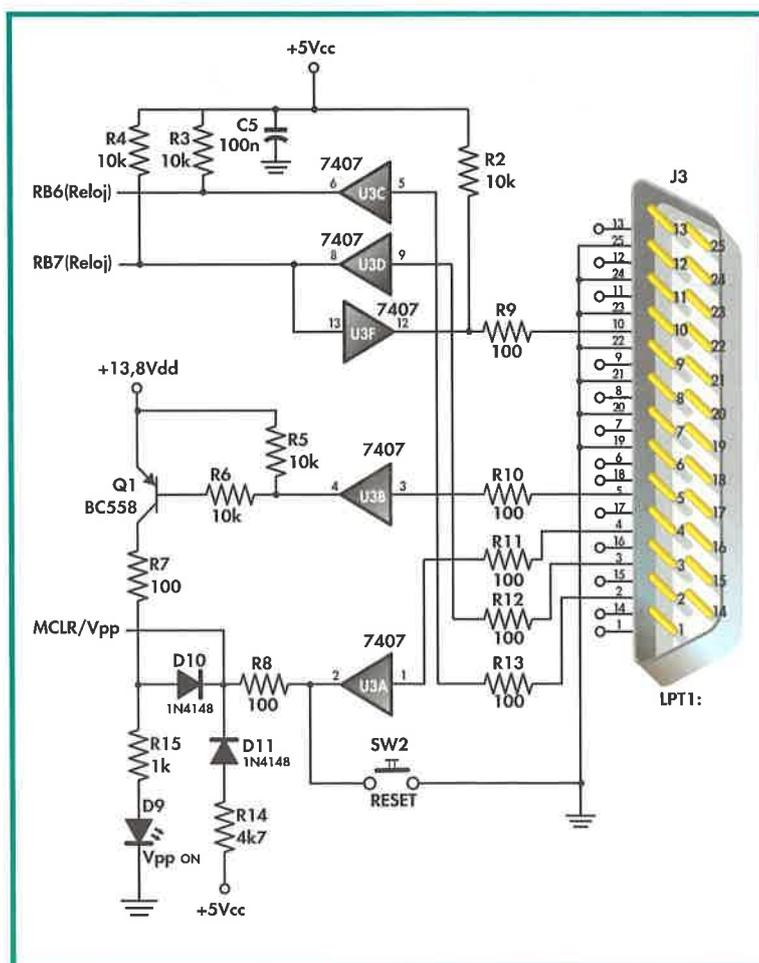
Come possiamo osservare questo circuito programmatore utilizza tensioni da 5 e da 13,8 V. Abbiamo già visto che i 13,8 V si ottengono con lo stabilizzatore 7812 e i tre diodi rettificatori 1N4007.

Questa tensione è necessaria per iniziare il processo di scrittura del PIC, dato che per portare il microcontroller in modo scrittura, è necessario inviare un segnale di reset (sul piedino MCLR) di 13,8 V.

Per commutare questa tensione di 13,8 V utilizziamo un transistor PNP BC558; se nella base del transistor c'è tensione questo non conduce, e nel reset abbiamo 5 V che arrivano attraverso il diodo D11 e la resistenza R14. Quando tramite il piedino 5 della porta parallela si invia uno 0, il transistor si polarizza e lascia passare i 13,8 V al segnale di reset del PIC, attraverso il diodo D10. Quando arriva la tensione di 13,8 V il diodo LED D9 si illumina. Terminata la scrittura del PIC si annulla il segnale di 0 sul piedino 5 del connettore DB25 e si manda un segnale di reset di 5 V per mezzo del piedino 4 del connettore, in modo che il PIC rimanga programmato e possa eseguire il programma.

Per introdurre il codice macchina nella memoria del microcontroller, si utilizzano i segnali provenienti dai pin 2 e 3 del connettore DB25, i quali vanno sui pin RB6 e RB7, rispettivamente. La memorizzazione si realizza bit a bit. Tramite la linea di R6 si invia un segnale periodico, facendo corrispondere ad ogni impulso positivo il valore del bit presente sulla linea che va a R7. Durante la lettura della

memoria di programma del PIC, il PC invia una sequenza di uno e zero attraverso la linea di R6 e contemporaneamente il PIC risponde inviando i bit tramite la linea di RB7 al piedino 10 del connettore. Tutte queste fun-



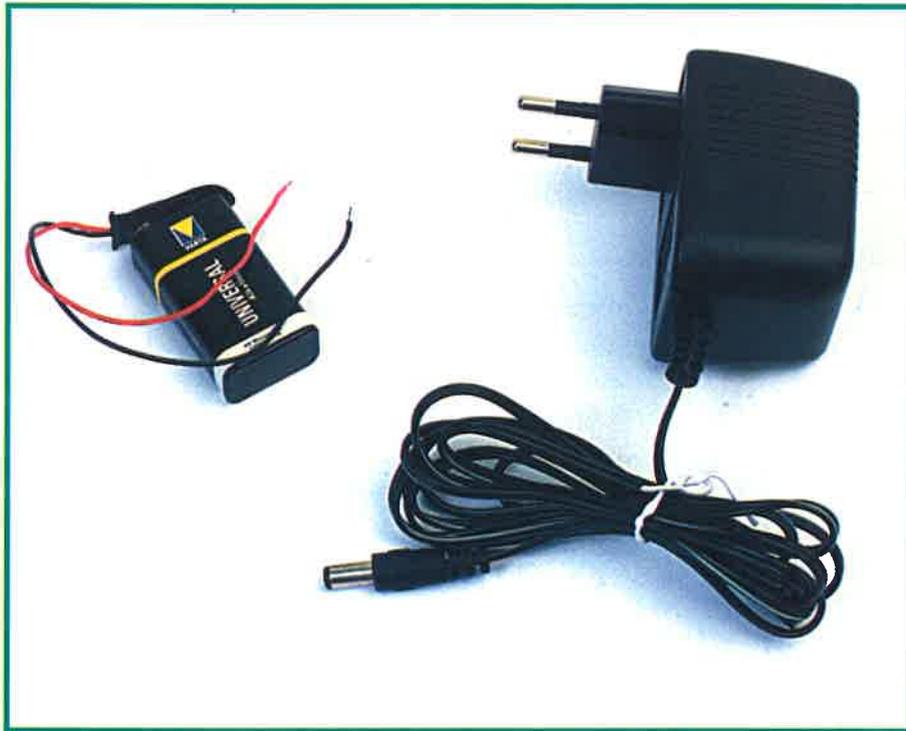
Schema impiegato per la programmazione del PIC.

zioni sono realizzate tramite il software di programmazione corrispondente, eseguito nel PC.

ALIMENTAZIONE DEL ROBOT

Come abbiamo già spiegato, l'alimentatore del robot è predisposto per ricevere sia tensione con-

tinua che alternata, tramite la quale alimenteremo tutto il robot. Per fornire tensione continua utilizzeremo le pile, mentre per la tensione alternata un trasformatore. Il trasformatore che utilizzeremo per alimentare questa scheda dovrà essere di 12 VAC.



Quando programmeremo il PIC è raccomandabile utilizzare sempre il trasformatore, dato che il consumo di questo procedimento è elevato, e potrebbero consumarsi le pile rapidamente.

In realtà mentre stiamo provando il programma sulla scheda, o non abbiamo ancora la versione definitiva, è consigliabile utilizzare sempre il trasformatore, per risparmiare le pile, e assicurarci una sufficiente quantità di energia.

Dopo che il microcontroller è stato programmato, alimenteremo il robot con le pile, in questo modo si potrà muovere liberamente, senza essere legato ad un cavo.



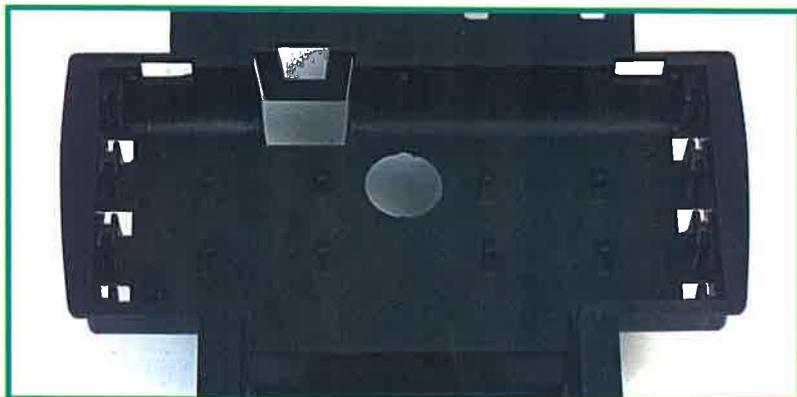
Un'alternativa alle pile convenzionali, consiste nell'uso di batterie ricaricabili.

Le batterie più comuni sono quelle al nichelcadmio, le quali possono essere ricaricate moltissime volte senza che si danneggino. Per le applicazioni che oltre all'autonomia richiedono anche elevate prestazioni, le batterie ideali sono quelle al piombo, che sono ugualmente ricaricabili, ma forniscono maggiore corrente.

MODULO DI CONTROLLO

PL 24

Montaggio passo a passo



Prepariamo ora il contenitore delle pile da 1,5 V del telaio. Collocheremo in questa sede sei pile in serie che ci forniranno un totale di 9 V, che utilizzeremo per alimentare i motori di Monty, e per ottenere la tensione necessaria per la programmazione del PIC.

Per fare in modo che le pile facciano contatto e possano essere collegate in serie, installeremo le sei placchette di connessione.

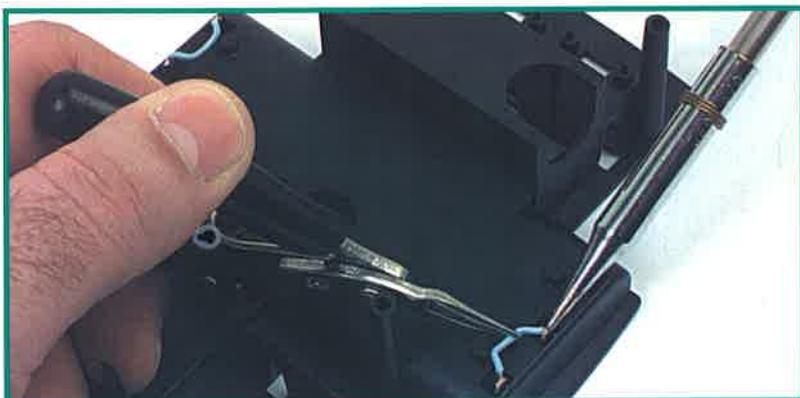


Dovremo introdurre in modo che l'aletta mobile della connessione rimanga orientata verso l'interno del contenitore. Inoltre dovremo schiacciare un po' questa aletta per facilitare l'inserzione delle pile.

Dovremo introdurre i contatti completamente, in modo che sporgano 6 terminali dalla parte superiore del telaio. Utilizzeremo questi terminali per saldare i cavi e collegare le pile in serie.



Montaggio passo a passo



Per realizzare i ponticelli utilizzeremo i cavi azzurri. Li taglieremo, li pelereemo e ne stagneremo i due estremi. Inoltre stagneremo anche i sei terminali superiori dei contatti delle pile. Realizzeremo i due ponticelli come mostrato nella figura.

Ora salderemo, con lo stesso procedimento, un filo rosso e uno nero, che sono quelli che forniranno all'elettronica della scheda, i 9 V. Il filo rosso corrisponderà al terminale positivo delle pile e verrà saldato nella parte superiore destra. Il filo nero corrisponderà al terminale negativo e si salderà in basso a sinistra.



Ora possiamo inserire le pile. Ne introdurremo due in ogni fila, e dovremo far pressione sino a che saranno perfettamente inserite.

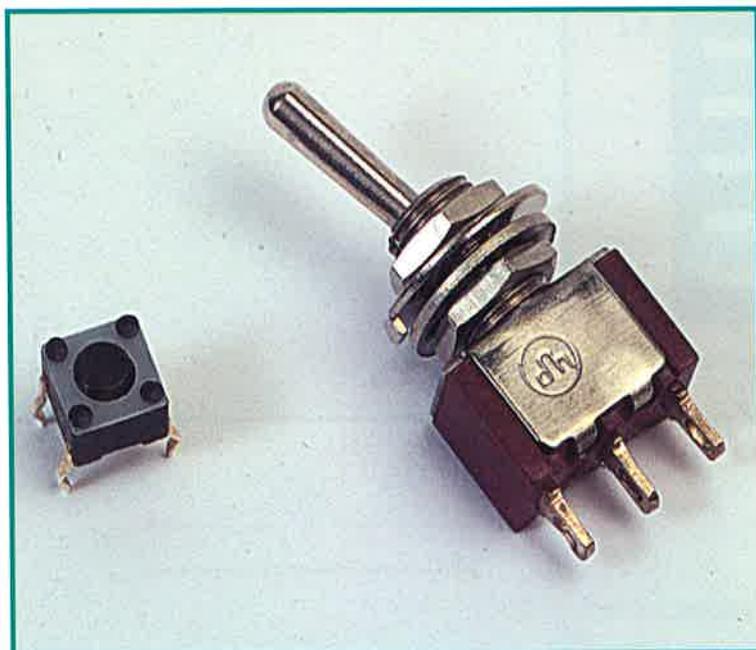
La sequenza in cui dobbiamo introdurre le pile è la seguente: nella fila superiore e in quella inferiore vengono montate due pile con verso + - + -. Nella fila intermedia la posizione delle pile sarà l'opposto: - + - +. La polarità delle pile è segnalata nella sede e deve coincidere con quella mostrata nell'immagine. Una volta introdotte le pile possiamo montare il coperchio.



INGRESSI DIGITALI

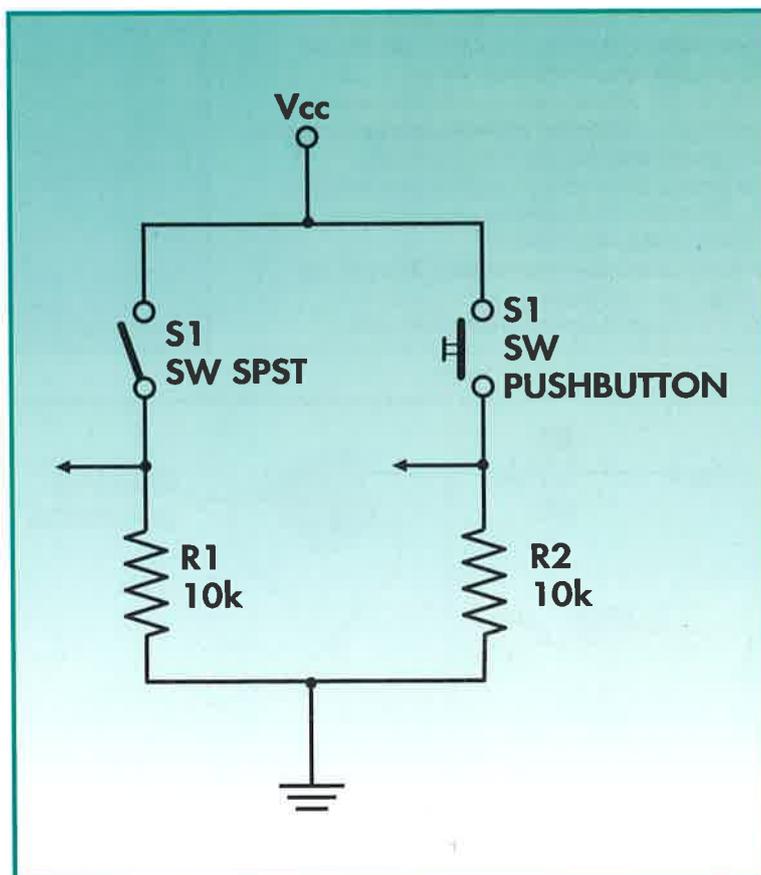
Gli ingressi digitali sono quelli che presentano solo due stati logici, che possiamo identificare come "1" o "0".

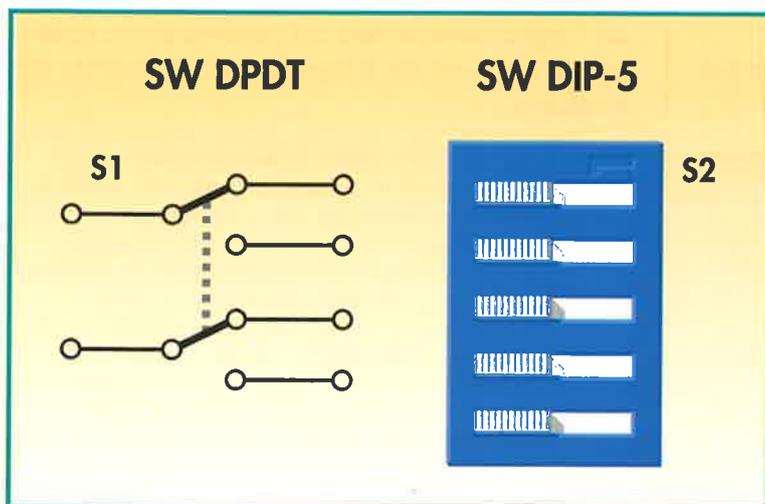
Un 1 logico corrisponderà alla presenza di una determinata tensione, ed uno 0 logico segnalerà l'assenza della medesima.



I dispositivi d'ingresso digitale più comuni sono gli interruttori e i pulsanti. Gli interruttori hanno un dispositivo meccanico, grazie al quale mantengono la loro ultima posizione. In questo modo, se mettiamo l'interruttore a "1", la sua uscita rimarrà in questo stato sino a quando non lo cambieremo a "0". Al contrario un pulsante ha uno stato logico permanente quando si trova a riposo, il quale potrà essere "1" o "0", questo dipende dallo schema elettronico impiegato. Lo stato della sua uscita cambia solo durante l'intervallo di tempo in cui viene attivato.

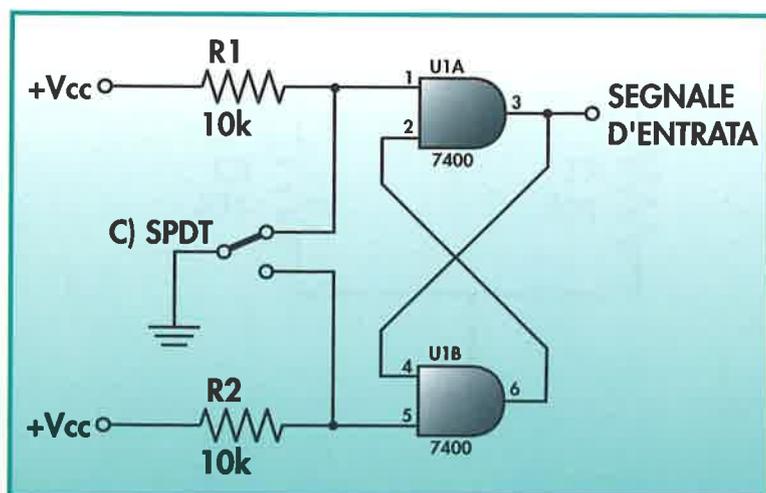
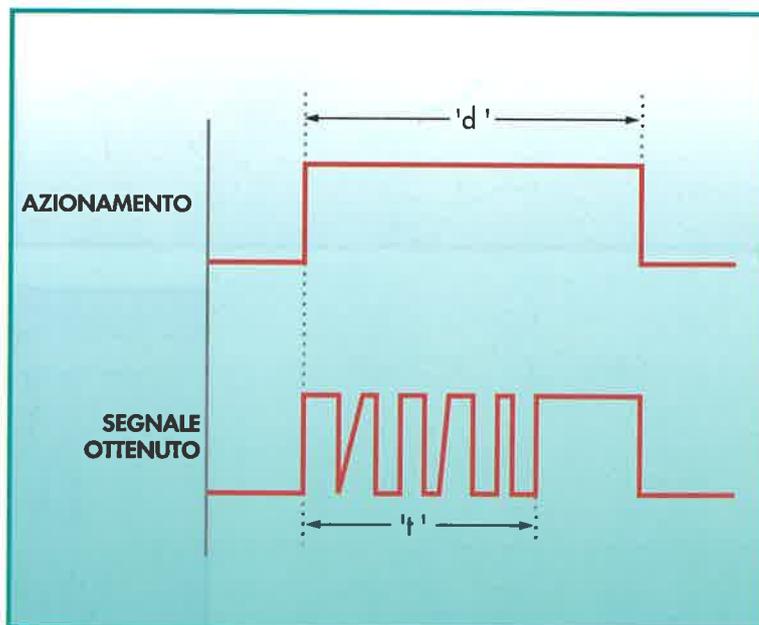
Questi sono gli schemi che dobbiamo realizzare per utilizzare un interruttore o un pulsante. In funzione del posizionamento e collegamento della resistenza, potremo ottenere all'attivarsi: un "1" o uno "0". Nello schema della figura, nella situazione di riposo entrambe le uscite sono a "0", quindi, quando si attiverà l'interruttore o il pulsante otterremo un "1" logico. I pulsanti e gli interruttori non sono necessariamente costituiti da un solo contatto; esistono commutatori e interruttori che alla loro attivazione chiudono numerosi contatti. Inoltre ci sono interruttori che chiudono fra i loro estremi diverse uscite, a seconda della posizione in cui si trovano. Per gli interruttori disponiamo inoltre di pacchetti di microinterruttori, come quello mostrato nell'immagine, che permettono di disporre in un unico componente di diversi ingressi digitali.





Sia gli interruttori che i pulsanti, sono periferiche d'ingresso di tipo meccanico. Tutte le periferiche ad ingresso meccanico presentano un fenomeno conosciuto come effetto rimbalzo.

Questo effetto si manifesta al cambiamento dello stato logico della periferica, quando si produce uno stato transitorio, sino al nuovo stato, in cui l'uscita è arbitraria e varia da 0 a 1. Questo succede perché in un periodo di tempo molto piccolo vengono prodotte diverse chiusure e aperture delle placche metalliche di cui sono costituite le periferiche meccaniche. La durata dell'effetto rimbalzo è molto piccola, però, siccome il tempo di elaborazione del microcontroller è ancora minore, se non controlliamo i rimbalzi, i programmi del PIC non funzioneranno correttamente. Per contrastare l'effetto dei rimbalzi esistono soluzioni hardware e software. Le soluzioni software consistono nel fare una temporizzazione con il PIC, ogni volta che si rivela un cambio di uscita di un interruttore, per aspettare che si stabilizzi. Le soluzioni hardware, cercano di annullare l'effetto rimbalzo all'uscita dell'interruttore, in modo che vari solo da 1 a 0 o da 0 a 1 una volta avvenuta la commutazione.



Nell'immagine si vede un esempio di circuito antirimbalo. Questo circuito si basa sull'impiego di un flip-flop RS con memoria, in modo che il segnale, passando a 1 o a 0, produca un'uscita "1" o "0" fissa sin dal primo momento.

MODULO DI INGRESSI-USCITE

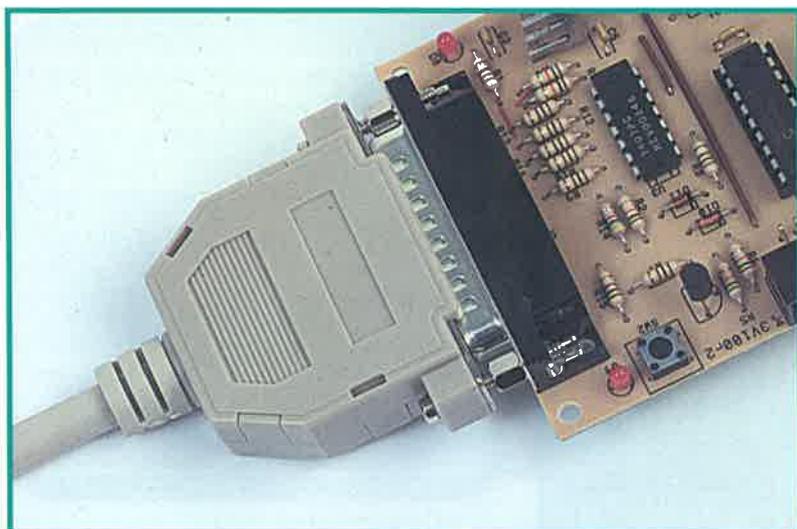
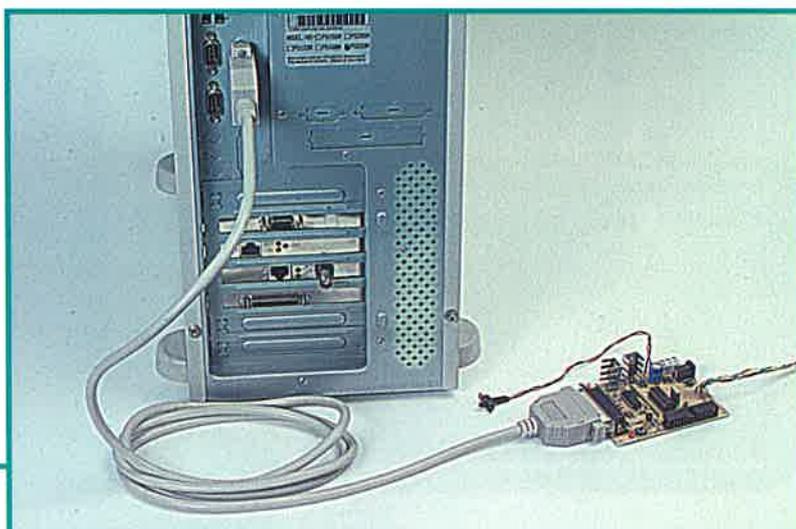
PL 26

Montaggio passo a passo

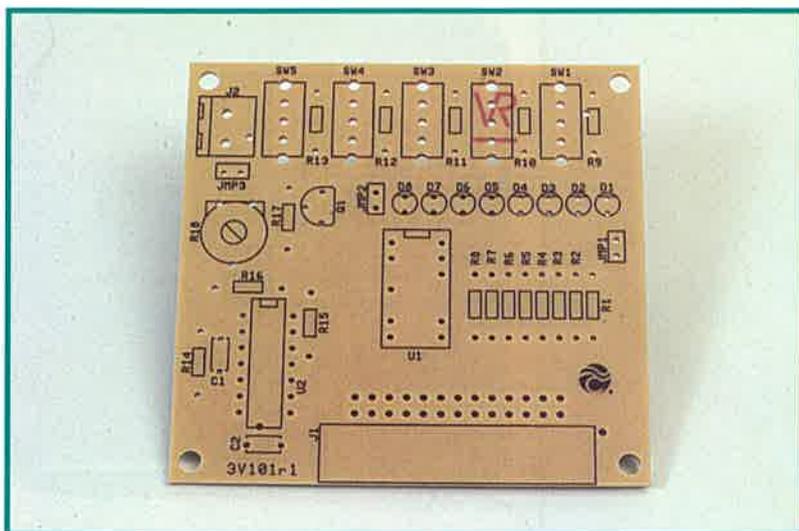


La scheda di controllo in cui è montato il microcontroller, servirà per la programmazione del medesimo. Per realizzare questo lavoro dobbiamo utilizzare un cavo parallelo con connettore DB25, come quello che si vede nell'immagine. Il cavo avrà ad un estremo un connettore femmina, e all'altro uno maschio. Questo tipo di cavo è standard e può essere acquistato in tutti i negozi di informatica e di elettronica.

Dobbiamo collegare il connettore femmina con il connettore maschio DB25 che è stato saldato sulla scheda di controllo (13). L'estremo maschio del cavo si collegherà all'uscita LPT1 del computer. Questa uscita è quella della porta parallela del PC, ed è la stessa che utilizziamo per collegare la stampante e alcuni scanner.

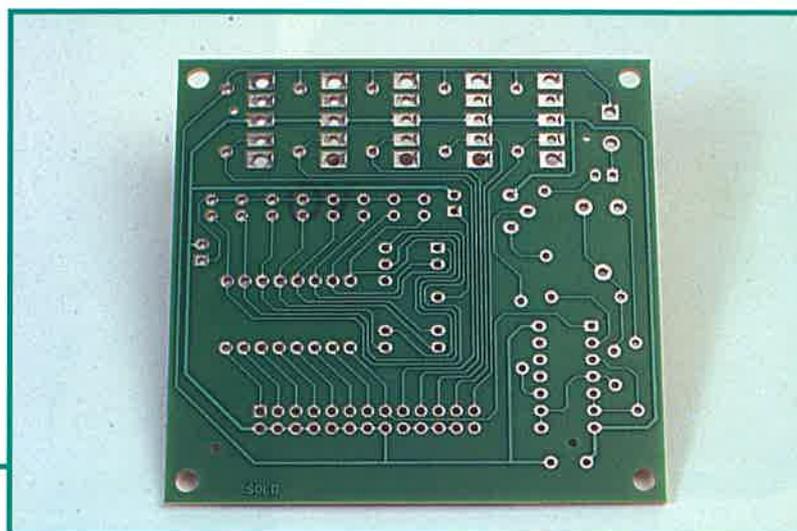


Presteremo molta attenzione nel collegare il cavo alla scheda, per non forzare questo connettore, dato che con l'uso potrebbero rovinarsi alcune saldature. Quando memorizzeremo diversi programmi o non, dovremo togliere il cavo dalla scheda per un determinato tempo; raccomandiamo di lasciare il cavo avvitato alla scheda come si vede nell'immagine.

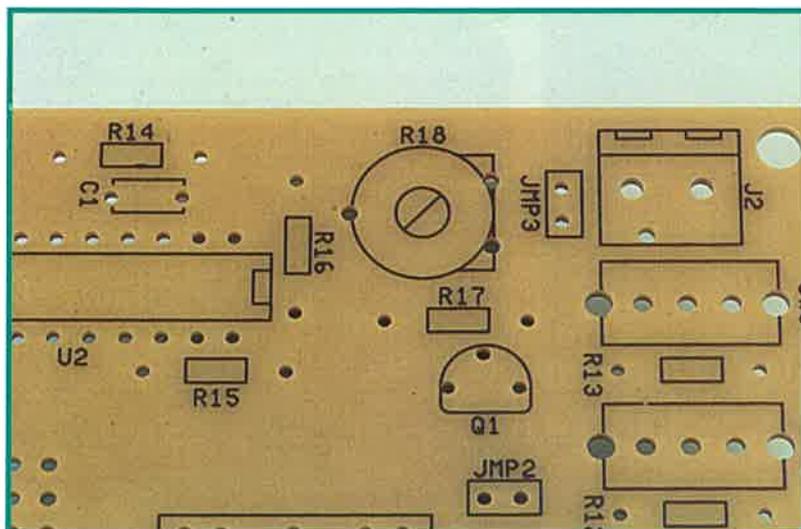


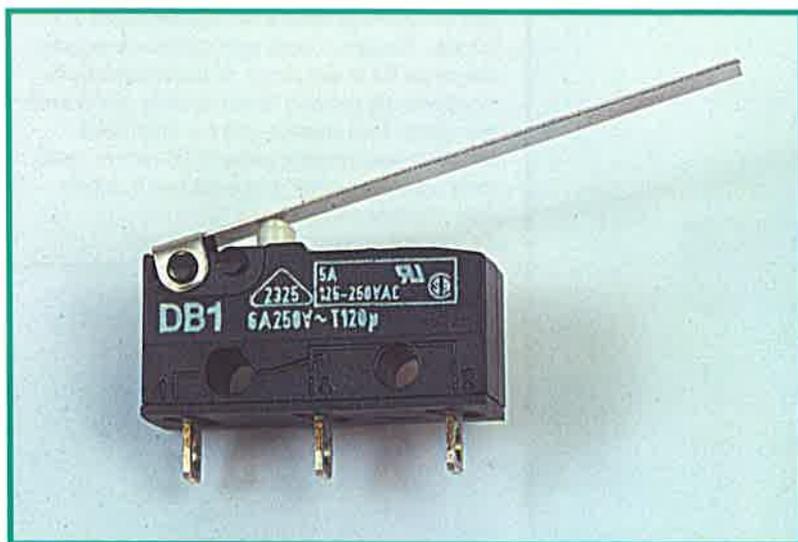
Questo circuito stampato è il prossimo che monteremo. Lo chiameremo Scheda di Ingressi e Uscite. Questa scheda ci servirà per imparare a programmare il microcontroller, dato che potremo valutare i programmi con periferiche reali, con ingressi ad interruttori, e uscite a display. Come potremo vedere, il metodo più efficace per assicurarci della validità di un programma, consiste nel provarlo attraverso questa scheda. La valutazione dei programmi tramite il simulatore del PC, è molto meno affidabile, perché, non essendo in tempo reale, non tiene conto dei fenomeni fisici dei componenti elettronici reali, come l'effetto rimbalzo degli interruttori (fenomeno di cui parleremo).

Questa scheda, così come la scheda di controllo, è un circuito stampato a singola faccia, dato che ha le piste su un solo lato. Per montare dei componenti su questa scheda bisogna seguire gli stessi procedimenti e consigli che abbiamo applicato per il montaggio della scheda di controllo.



La Scheda di Ingressi-Uscite, inoltre, è serigrafata sul lato superiore, in questo modo consultando la descrizione presente sulla propria scheda sapremo dove dovrà essere collocato ogni componente, così come il verso, nel caso si tratti di un elemento con polarità.

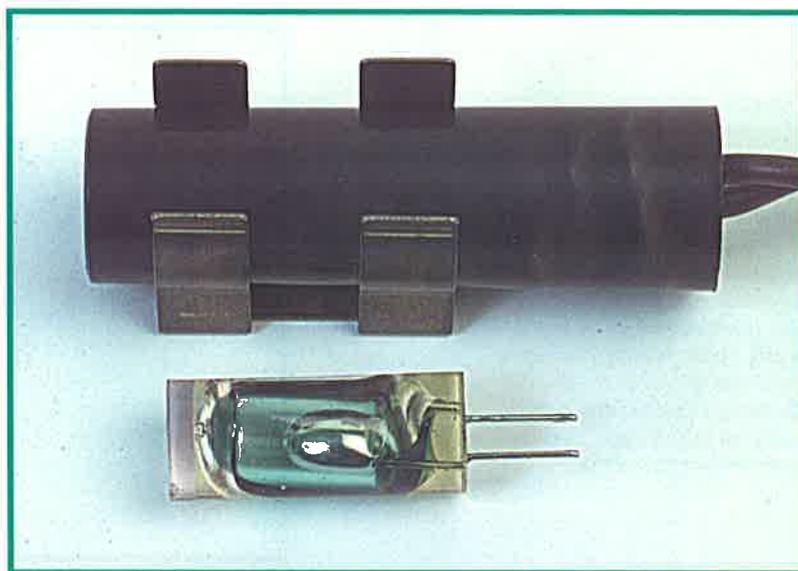




Oltre agli interruttori e ai pulsanti, esistono altri tipi di sensori meccanici adatti all'uso del mondo della microrobotica. Qualunque sensore di tipo meccanico che utilizzeremo, presenterà sempre l'effetto rimbalzo.

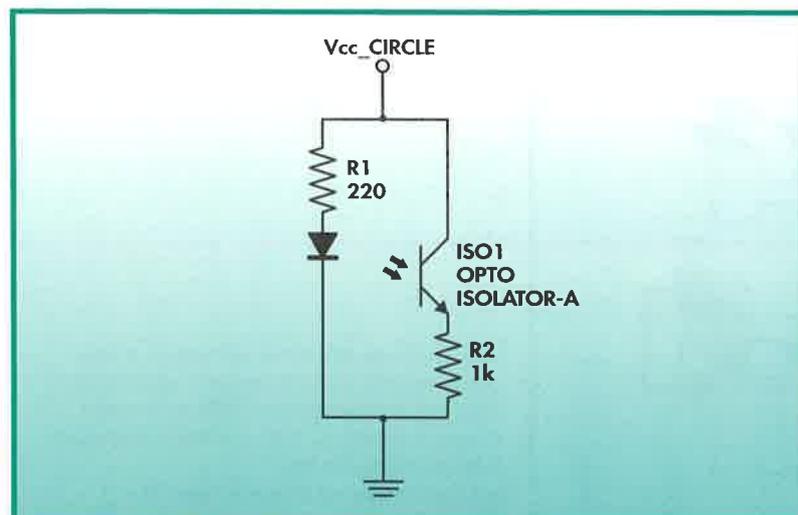
Nell'immagine è mostrato un sensore di ingresso conosciuto come fine corsa o bumper. Questo interruttore dispone di una lamina metallica, che può avere diverse lunghezze a seconda del modello, e che attiva il sensore ogni volta che viene premuta. In questo modo potremo utilizzare i fine corsa per sapere quando Monty va a sbattere contro un oggetto, e in quale posizione si trova l'ostacolo.

Un altro modello di sensore digitale è quello al mercurio. Questi sensori utilizzano il movimento del mercurio allo stato liquido per chiudere o aprire dei contatti. Possiamo utilizzare questi sensori in un robot, per rilevare l'inclinazione della superficie sulla quale si sta muovendo, in modo da poter dare l'ordine di retromarcia nel caso che questa inclinazione rischi di far ribaltare il robot. **AVVERTENZA:** il mercurio è una sostanza tossica per cui questo tipo di dispositivo non deve essere né tagliato, né aperto.

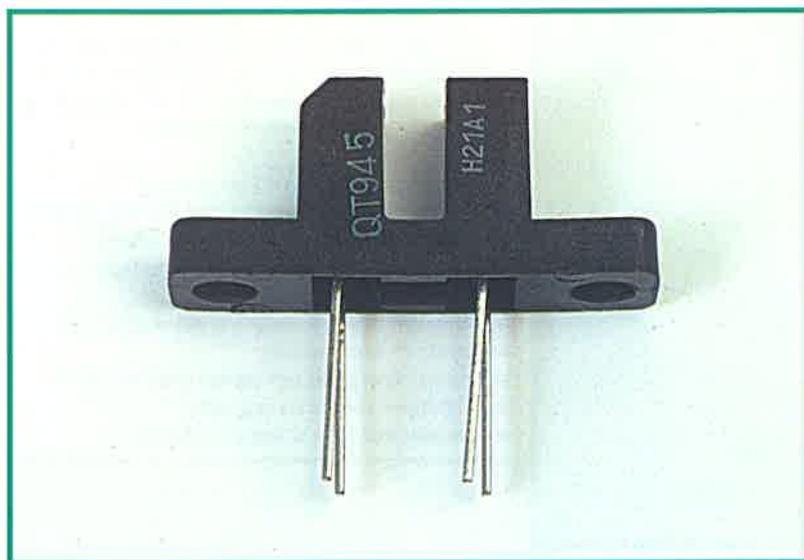


Sensori ottici

I sensori ottici sono sensori digitali che utilizzano un raggio di luce infrarossa per il loro funzionamento. Si basano sull'uso di un fotodiode e di un fototransistor, che sono componenti elettronici capaci di modificare il loro comportamento in funzione della quantità di luce che incide sopra di essi.

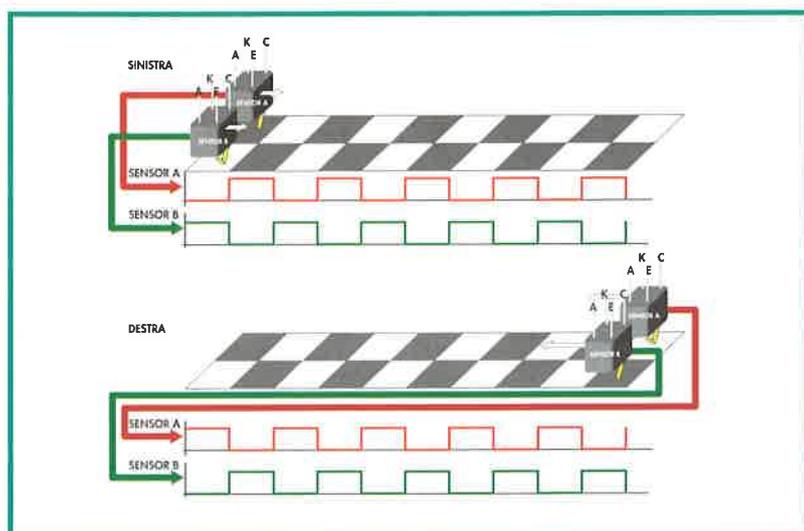
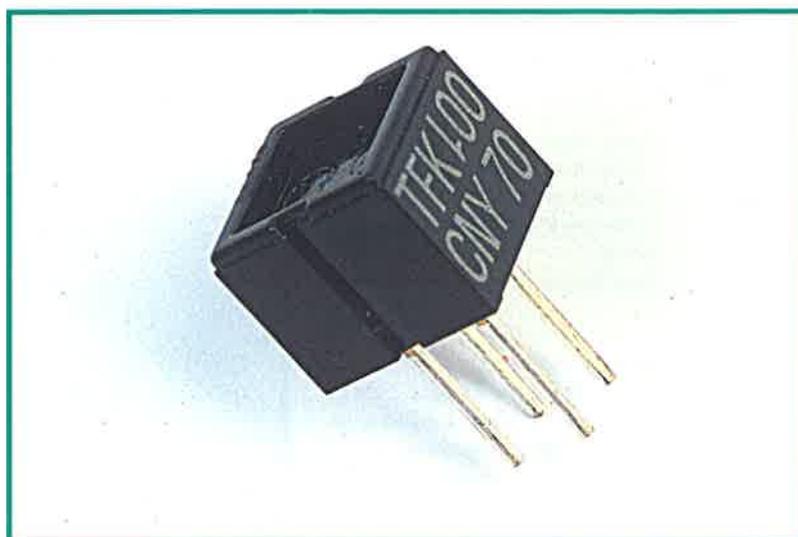


I sensori ottici più comuni utilizzano un fotodiode come fonte di emissione del raggio di luce, e un fototransistor come ricevitore di detta luce. Il fototransistor è un tipo di transistor che polarizza la sua base quando c'è presenza di luce, quando invece non riceve luce, non lascia circolare corrente tra collettore ed emettitore.



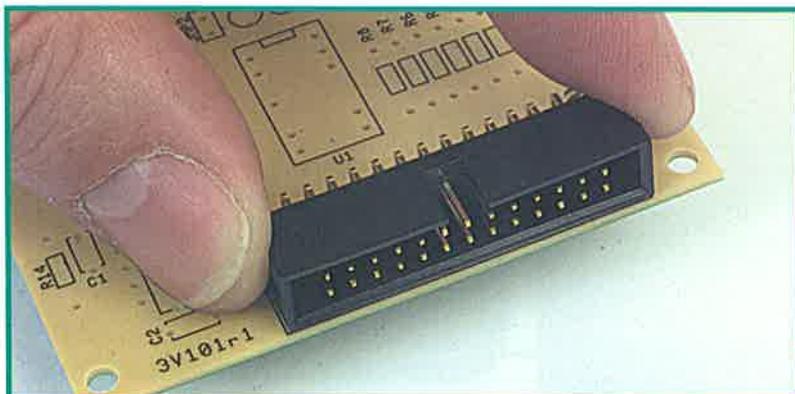
Nell'immagine si mostra un sensore ottico a forcella. Questo sensore può rilevare un oggetto che passa tra le sue pareti. In questo modo, se montiamo un pezzo a forma di stella sull'asse di una ruota, e un sensore ottico a interruzione che conti quante volte passano gli estremi della stella tra le sue pareti, potremo fare il calcolo della distanza percorsa dal robot, calcolare la sua velocità, ecc.

Il sensore dell'immagine è un sensore ottico a riflessione. Emette un raggio di luce con un determinato grado di inclinazione. Se questa luce rimbalza sopra una superficie bianca e ritorna al fototransistor, l'uscita del sensore andrà a "1". Al contrario, se la luce non rimbalza, o la superficie è nera (quindi assorbe la luce) il fototransistor non si polarizza, e all'uscita del sensore avremo uno "0". Con questo principio di funzionamento si può intuire come faremo a far sì che il robot segua la traiettoria di una linea nera disegnata sul pavimento, oppure che riconosca il verso della curva, gli spostamenti, ecc.



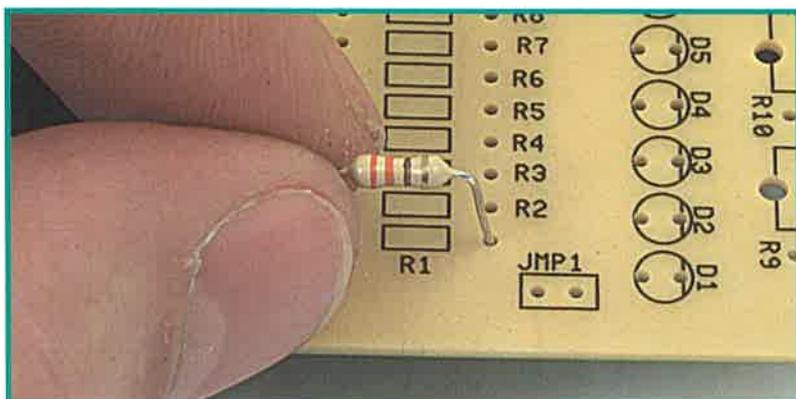
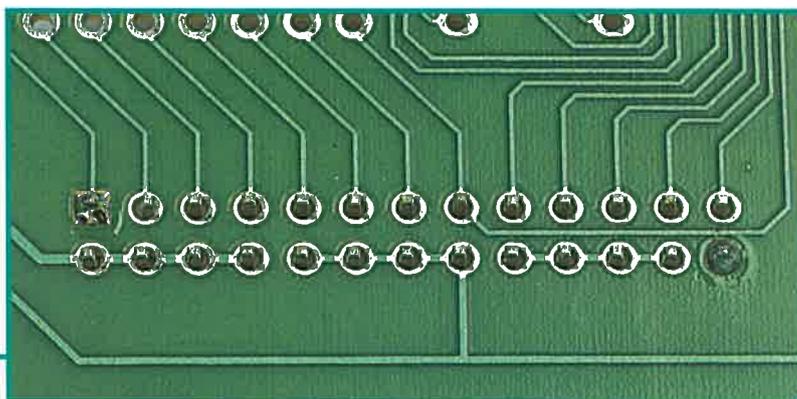
Un'importante differenza dei sensori di tipo ottico, rispetto a quelli meccanici, consiste nel fatto che non presentano l'effetto rimbalzo. Questo è dovuto al fatto che non ci sono parti meccaniche coinvolte nel funzionamento di questi dispositivi.

Montaggio passo a passo



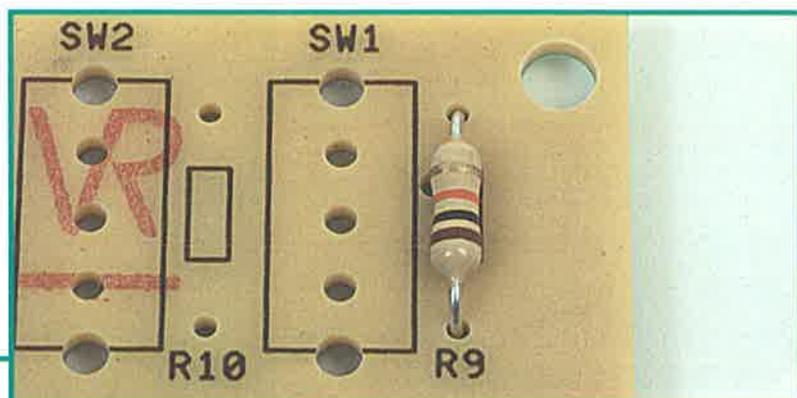
Collegeremo ora il connettore a 26 vie a 90°, che sarà il connettore di accesso alla scheda di ingresso e uscita mediante il PIC-BUS. Questo connettore sarà montato in J1, nel verso mostrato dall'immagine, a filo della scheda. Quanto maggiore appoggio avrà il connettore, meno soffriranno le saldature introducendo ed estraendo il cavo del PIC-BUS.

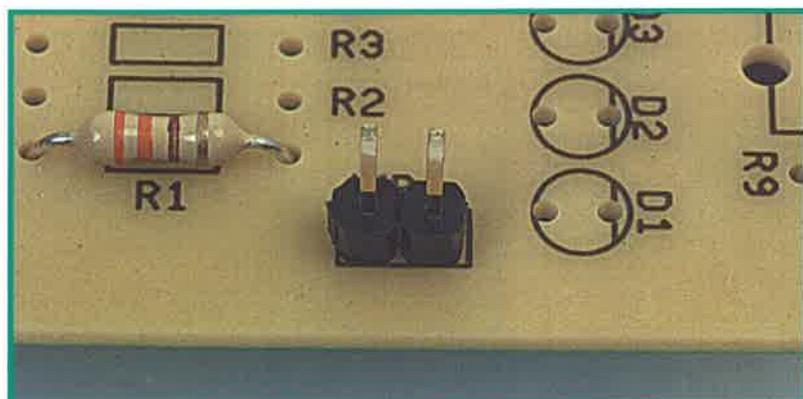
Per saldare questo connettore inizieremo dai due piedini posti agli angoli del medesimo, i quali devono essere agli estremi opposti. Una volta realizzate queste due prime saldature, verificheremo che il connettore sia ben inserito nella scheda, quindi proseguiremo con le altre.



Il passo successivo sarà montare la resistenza da 330Ω. Questa resistenza si identifica per i colori arancio-arancio-marrone. La sua posizione è R1 e non ha polarità.

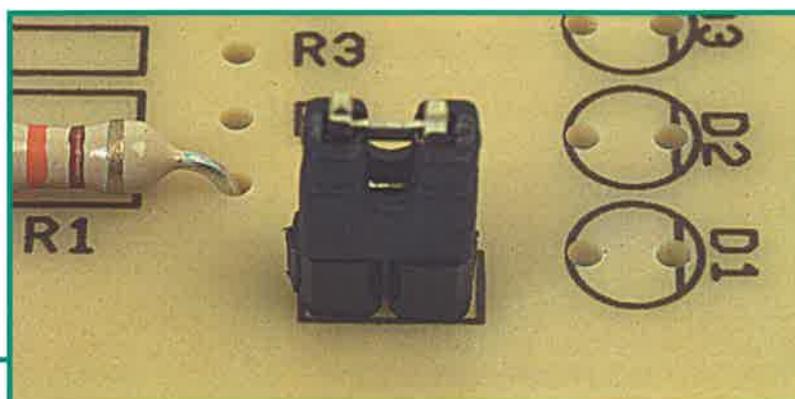
Ora salderemo la resistenza R9 il cui valore è 10 KΩ. Questa è identificata dai colori marrone - nero - arancio. Si raccomanda di montare la banda dorata di entrambe le resistenze nello stesso verso, per questioni di estetica.



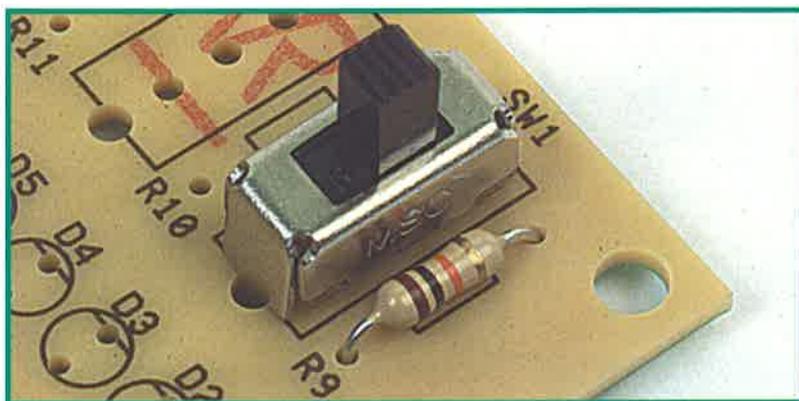


Dobbiamo saldare il connettore dritto a due vie in JMP1. La parte corta del connettore è quella che deve essere introdotta nella scheda e poi saldata. Dobbiamo assicurarci di saldare il connettore a due vie lasciando i pin perfettamente verticali rispetto alla scheda.

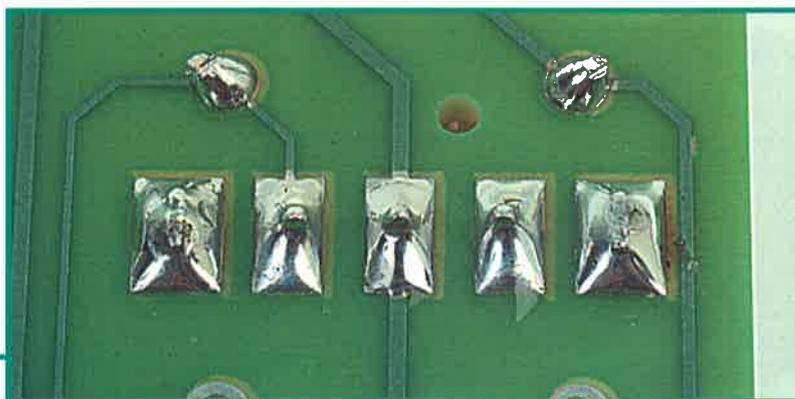
Il jumper è un componente che si utilizza per inserire o escludere parti di circuito. In questo modo se collochiamo il jumper su JMP1, metteremo in contatto la coppia di pin, se al contrario lo toglieremo, lasceremo la linea aperta.



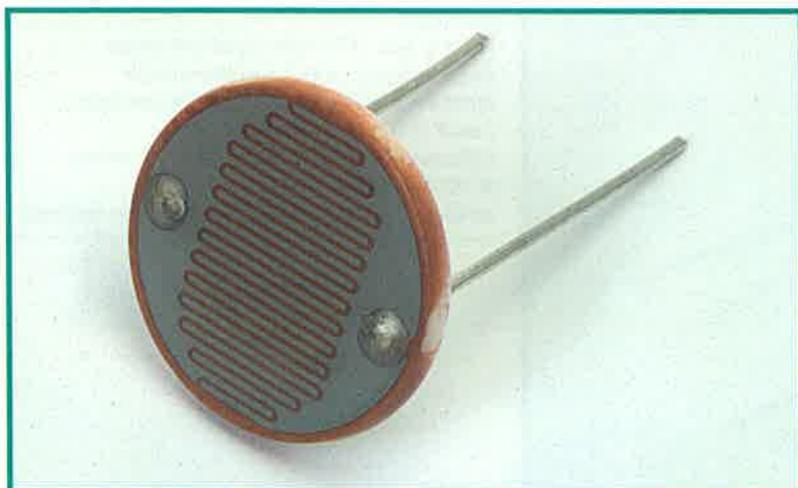
Monteremo ora il commutatore a slitta a due posizioni. Questo commutatore si introdurrà in SW1. Dovremo premere per fare incastrare perfettamente il commutatore alla scheda, in modo che rimanga ancorato alla medesima.



Salderemo i cinque piedini dell'interruttore. Di questi solo i tre centrali hanno una funzione elettrica, però saldando anche i due esterni, otterremo maggiore stabilità. Dovremo fondere stagno a sufficienza per riempire completamente i fori.



Consigli pratici



Ingressi analogici

Gli ingressi analogici possono avere un numero illimitato di valori fra due range. Oltre ai sensori digitali che abbiamo già visto, vi sono moltissimi altri sensori i cui segnali di uscita sono di tipo analogico. In seguito mostreremo alcuni esempi.

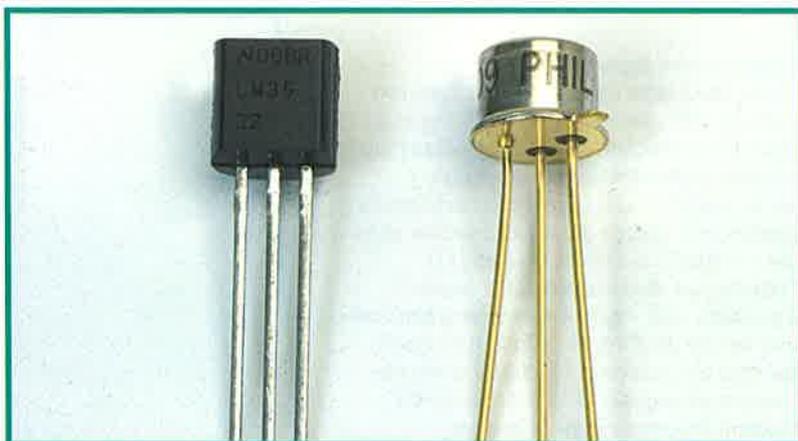
Cella LDR

La cella LDR mostrata nell'immagine, è un tipo di resistenza che varia la sua resistività in funzione della quantità di luce che incide su di essa. Con questa proprietà e un piccolo circuito di adattamento, possiamo misurare, ad esempio, la quantità di luce che c'è in un'abitazione.

Sensori di temperatura a giunzione di semiconduttore

Per misurare la temperatura sono molto usati i sensori a giunzione di semiconduttore, come i modelli LM35 o AD590.

Questi tipi di sensori modificano la tensione o la corrente alla loro uscita in funzione della temperatura. Sono sensori molto lineari. Senza dubbio, il nostro microcontroller, cervello del robot, è un microprocessore di tipo digitale che può solo lavorare con dati binari. Come faremo, quindi, per leggere i segnali analogici? Abbiamo due soluzioni:

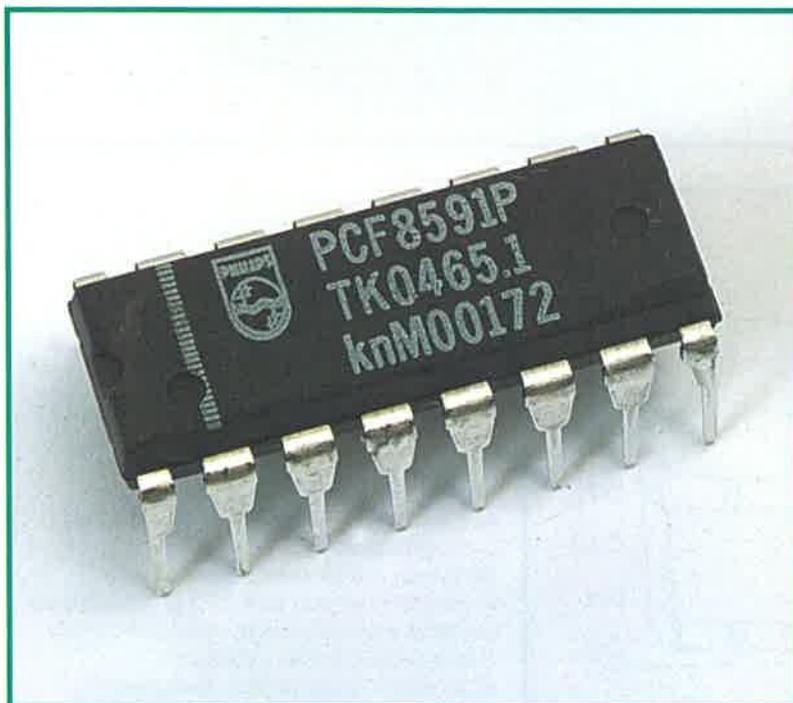


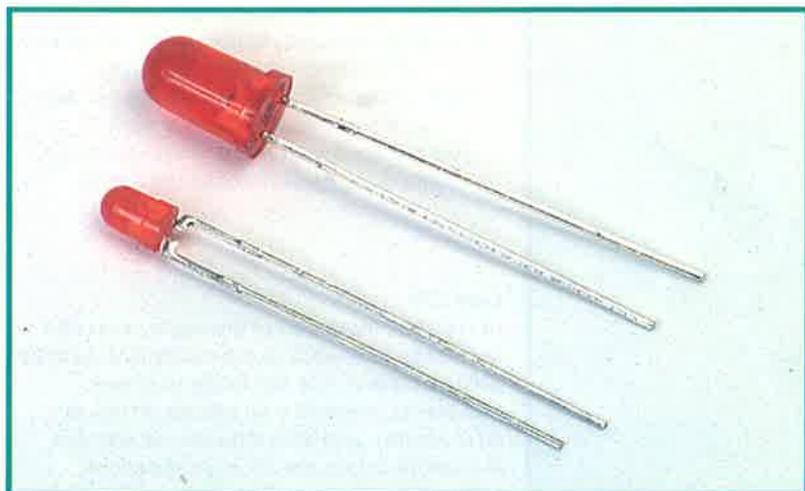
Convertitore analogico/digitale

Una forma di comunicazione mediante questo tipo di segnali, si può avere utilizzando convertitori analogico/digitali. Si tratta di chip che leggono il segnale analogico e lo trasformano in valore digitale (codice binario), secondo soglie prestabilite. Dato che l'uscita di questi chip è digitale la potremo inviare direttamente al PIC. Un'altra soluzione consiste nell'utilizzare dei circuiti esterni di condizionamento a soglia, per fare in modo che l'uscita dei sensori si converta in un livello "1" o "0", a seconda che venga raggiunta o meno la soglia limite per cui abbiamo progettato il circuito. È facile capire che con questa seconda soluzione non potremo leggere il dato reale dell'uscita del sensore.

Uscite digitali

Così come i segnali d'ingresso che arrivano al microcontroller sono di natura digitale, i segnali di uscita che questo genera sono anch'essi digitali. Si tratta di segnali che variano da 5 a 0 V con un'intensità massima di 20 mA per piedino. Con questa intensità potremo attivare direttamente alcuni sensori di uscita digitale; per altri avremo bisogno di aumentare la potenza.



**Diodi LED**

I carichi tipo LED, come quello mostrato nell'immagine, possono essere attivati direttamente dai piedini del microcontroller. I diodi LED sono periferiche molto utilizzate, dato che ci permettono di vedere lo stato logico dei segnali, senza necessità di utilizzare il tester per verificarli.

Display a sette segmenti

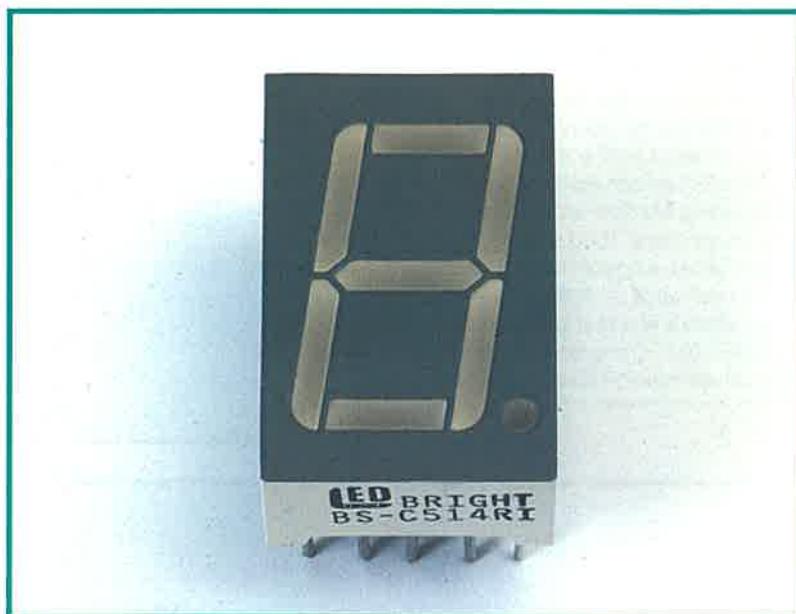
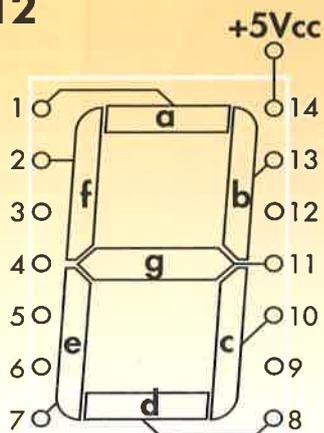
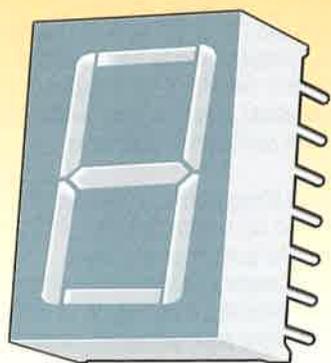
Un display a sette segmenti è un dispositivo costruito utilizzando dei diodi LED, con una disposizione speciale, in modo che illuminandoli possano rappresentare numeri digitali.

Perciò, anche i display possono essere attivati direttamente dai pin del microcontroller, visto che si tratta, in definitiva, di diodi LED.

Il display può essere ad anodo comune o a catodo comune, in funzione se il terminale comune dei diodi LED è l'anodo o il catodo.

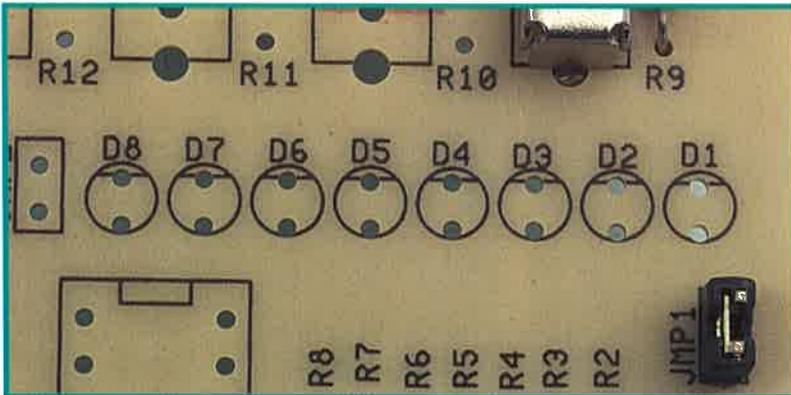
Nel caso che si tratti di un display di catodo comune, un segnale "1" che arriva dal PIC produrrà l'accensione del segmento.

Quindi se la configurazione del display è di anodo comune, un "1" del PIC manterrà un segmento del display spento e sarà necessario inviare uno "0" per accenderlo.

**TIL 312****Convertitore analogico/digitale**

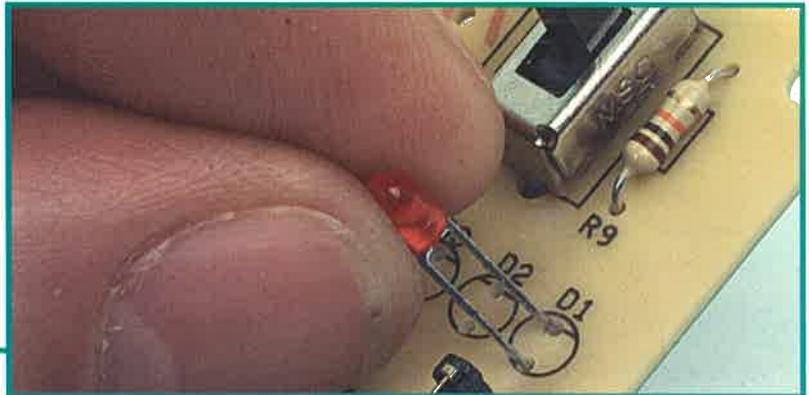
Nell'immagine vediamo un display ad anodo comune modello TIL312 che è alimentato con tensione positiva di +5 V al piedino 14 (anodo comune a tutti i segmenti). Per attivare ogni segmento individuale è necessario, pertanto, introdurre livello logico "0" sul piedino corrispondente.

Montaggio passo a passo



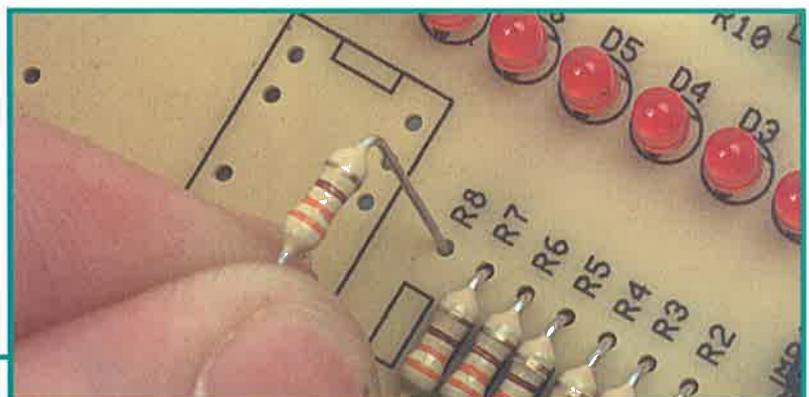
I diodi LED saranno collocati da D1 a D8. I diodi LED hanno polarità, e questa è segnata nella serigrafia della scheda. Nella scheda appare una circonferenza con una linea circoscritta in uno dei suoi bordi. Questa linea indica il catodo del diodo LED.

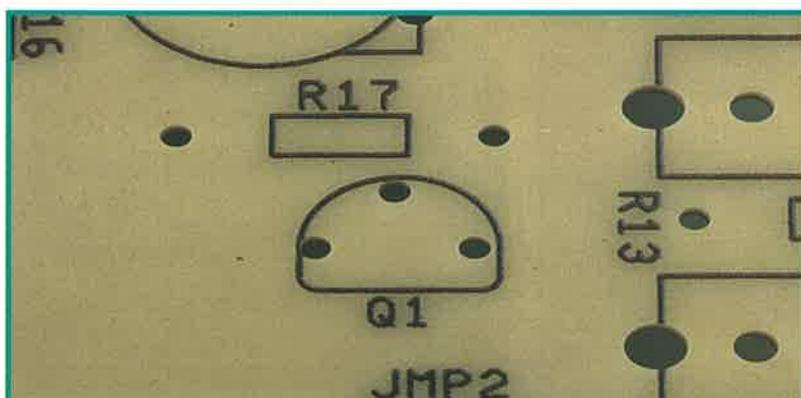
Il catodo rappresentato nel diodo LED è segnalato con il piedino più corto. Dobbiamo far coincidere i catodi dei diodi con i catodi segnati nella scheda. Tutti i diodi da D1 a D8, vengono inseriti nello stesso verso come si vede nell'immagine.



Dobbiamo fare in modo che tutti i diodi restino perpendicolari rispetto alla scheda, poiché, se sono storti avremo un brutto risultato estetico. Possiamo iniziare saldando solamente un piedino per ogni diodo, per correggere la sua posizione più facilmente nel caso fosse necessario.

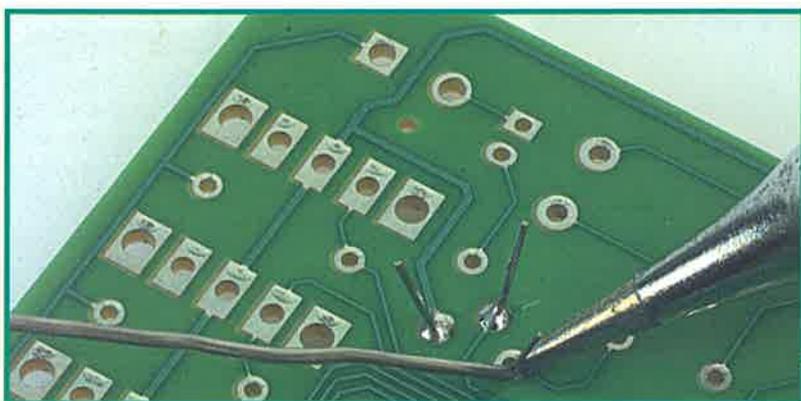
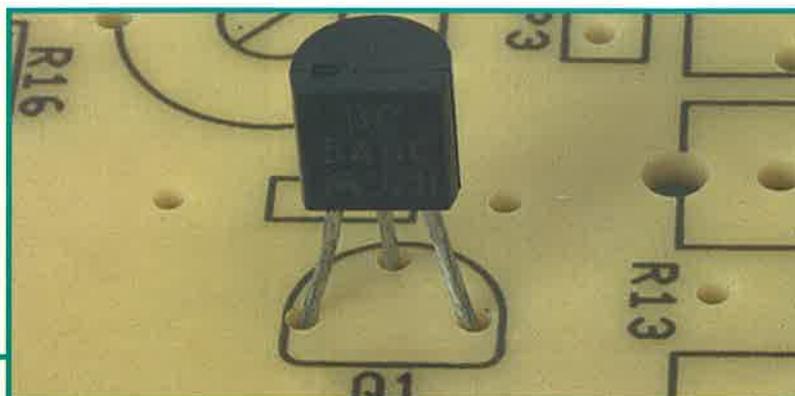
Adesso salderemo le resistenze da R2 a R8. Queste resistenze valgono 330Ω , e il loro codice colori è arancio-arancio-marrone. Collocheremo tutte le resistenze con la banda dorata, che indica la tolleranza, orientata nello stesso verso.





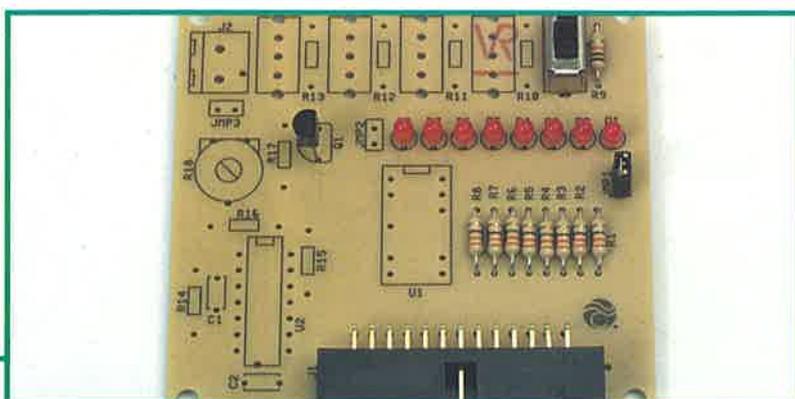
Il prossimo elemento da saldare sulla scheda è un transistor NPN, modello BC548. Questo transistor sarà montato in Q1. Il verso in cui deve essere inserito è segnalato sulla serigrafia della scheda, con un disegno a forma di mezza luna.

Dobbiamo inserire il transistor in modo che, visto da sopra, la sua forma coincida con la mezza luna disegnata sulla serigrafia. Il transistor non deve essere a filo della scheda, ma rialzato di alcuni millimetri dalla stessa.



Salderemo i piedini del transistor mettendo lo stagno da un lato e il saldatore dall'altro, secondo il procedimento che abbiamo imparato. Dopo aver saldato i piedini, taglieremo la rimanenza fino a lasciare un cono.

Questo è l'aspetto finale che deve avere la scheda con i componenti che abbiamo montato fino a questo momento. Verificate che tutti i componenti siano inseriti al loro posto e nel verso indicato.

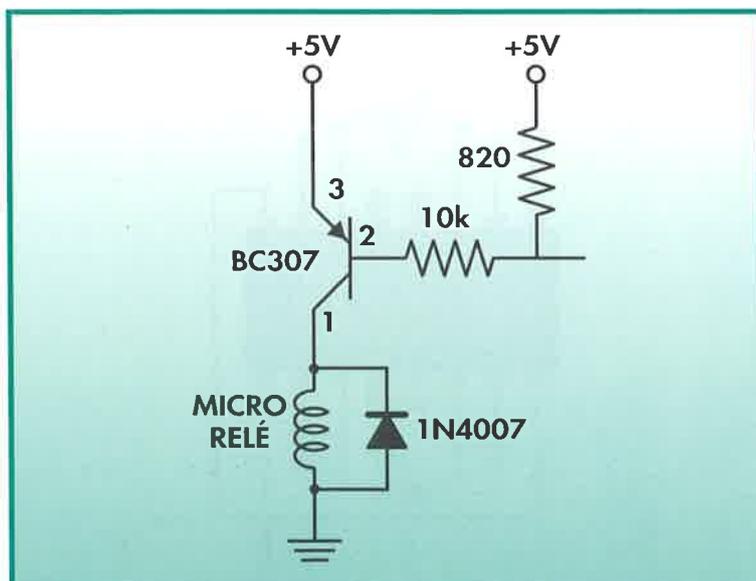


Consigli pratici



Oltre ai dispositivi digitali ottici di uscita, come i display o i led, disponiamo anche di dispositivi sonori. I sensori piezoelettrici, emettono un suono quando sono attivati da un '1' logico, e smettono di emettere quando ricevono uno '0'. Si possono attivare direttamente con il segnale di uscita di qualunque piedino del microcontroller PIC.

Oltre ai cicalini piezoelettrici, possiamo produrre suoni anche con un altoparlante. Un altoparlante avrà bisogno di un segnale periodico di '1' e '0' per far sì che vibri e produca un suono. A seconda della dimensione dell'altoparlante, e della potenza sonora desiderata, potrebbe essere necessario aggiungere degli amplificatori esterni; inoltre hanno il vantaggio di poter produrre suoni bassi o acuti, variando la frequenza del segnale periodico. Ne consegue una migliore resa sonora rispetto ai cicalini piezoelettrici.

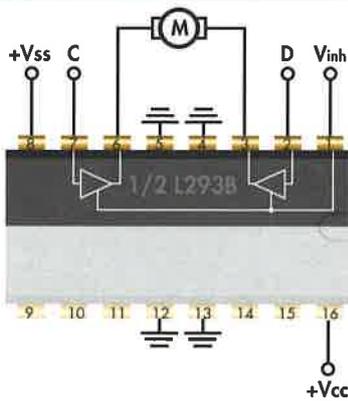
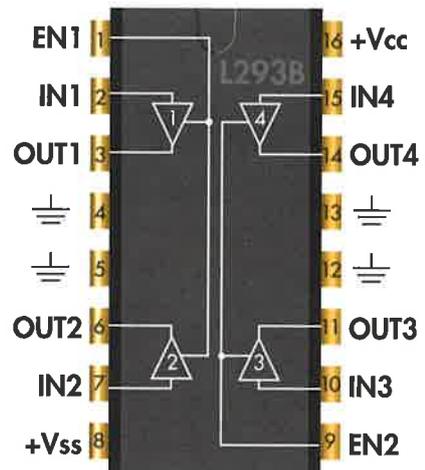


Quando si deve alimentare un'uscita che ha un alto consumo di corrente, si utilizza una tensione diversa dai 5 V di uscita del PIC. Possiamo utilizzare un relé i cui contatti permettono di lavorare con tensioni e consumi elevati. Lo schema ci mostra un possibile collegamento della bobina alla linea di uscita del PIC. In questo modo, isoliamo il segnale del PIC dal carico che dobbiamo attivare.



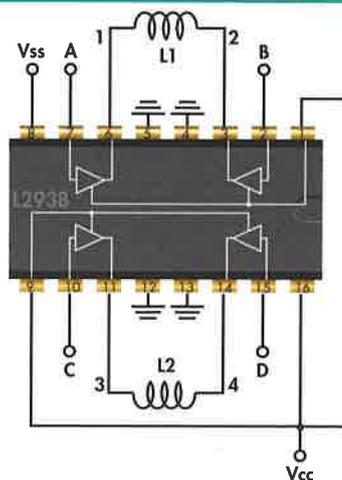
Delle periferiche tipiche, impiegate in numerose applicazioni (ad es. in Monty), sono i motori. Nell'immagine abbiamo un motore a corrente continua (C.C.). Il microcontroller può pilotare la sua accensione, la sua fermata, così come la velocità e il verso del giro.

Per convertire i segnali logici del PIC in segnali di potenza, applicabili al motore, possiamo utilizzare un driver come l'L293B che può governare sino a due motori CC. Questo chip si controlla direttamente dal PIC e amplifica il segnale per poter attivare il motore. Inoltre si può impiegare il classico ponte ad H, formato da transistor e componenti discreti.

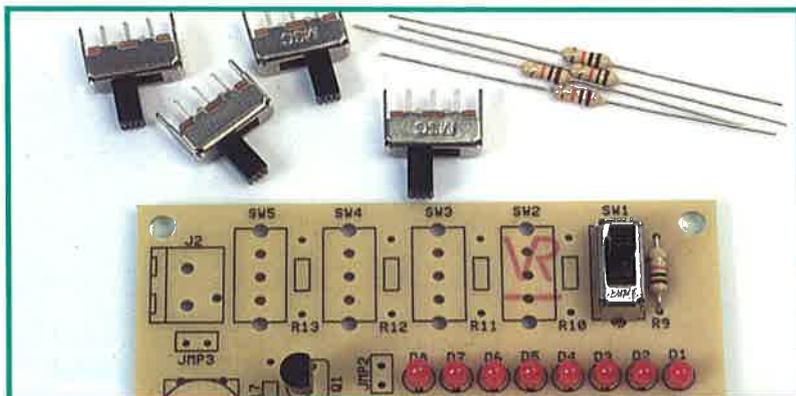


Se colleghiamo un motore CC come mostrato dall'immagine, possiamo controllare, utilizzando solamente due segnali del PIC, l'accensione e lo spegnimento del motore stesso, e anche il verso del giro. L'accensione e lo spegnimento si ottengono abilitando o meno il segnale di Enable del driver (V_{inh}). Per cambiare il verso del giro dobbiamo solo inviare la combinazione '1'-'0' oppure '0'-'1' ai piedini C e D. La tensione di alimentazione del motore si applica tramite l'ingresso V_{ss}.

Con questo driver inoltre, è possibile gestire dei motori passo a passo, collegando ognuna delle bobine nei due lati del driver. Tramite il PIC dovremo solo inviare al driver la sequenza appropriata dei codici binari per far sì che il motore giri a destra o a sinistra ad una determinata velocità.

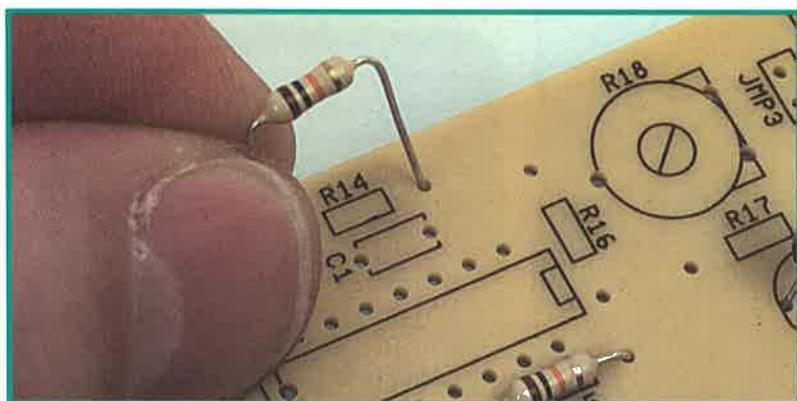
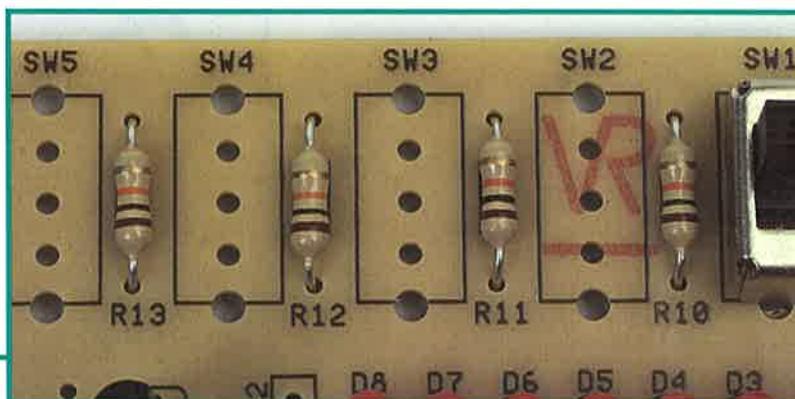


Montaggio passo a passo



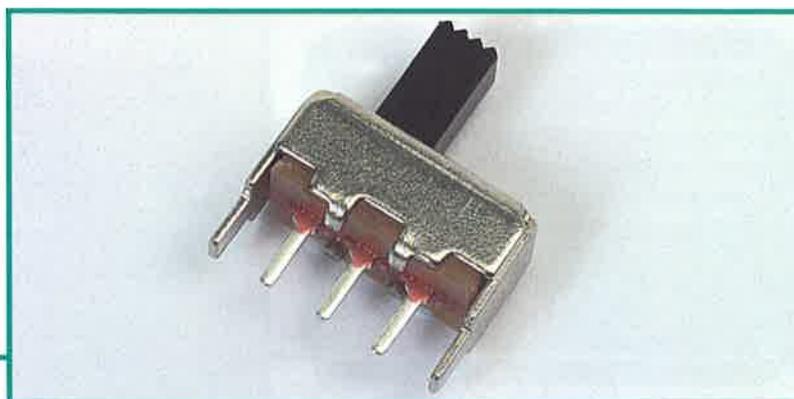
Procederemo a saldare il resto dei commutatori della scheda così come le resistenze da $10\text{ K}\Omega$ associate. I commutatori vanno da SW2 a SW5, e le resistenze da R10 a R13.

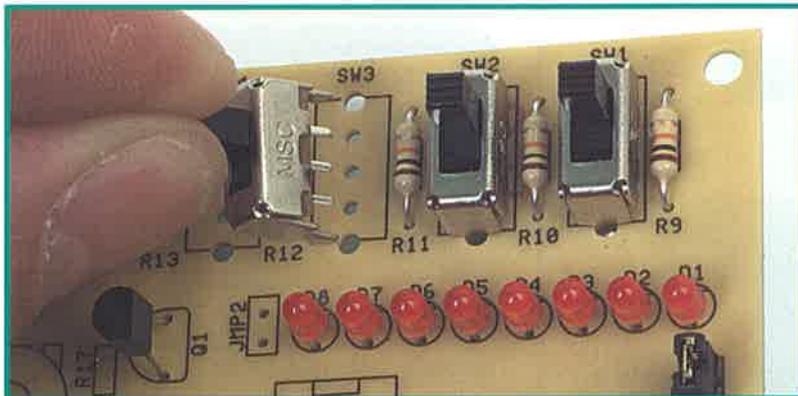
Prima monteremo le resistenze, dato che sono più basse dei commutatori e potremo lavorare meglio se le salderemo prima. Le resistenze hanno il codice colori marrone-nero-arancio, il loro valore è $10\text{ K}\Omega$ e occupano le posizioni da R10 a R13.



Salderemo anche le altre due resistenze da $10\text{ K}\Omega$ (stesso codice colori) che saranno inserite nelle posizioni R14 e R15.

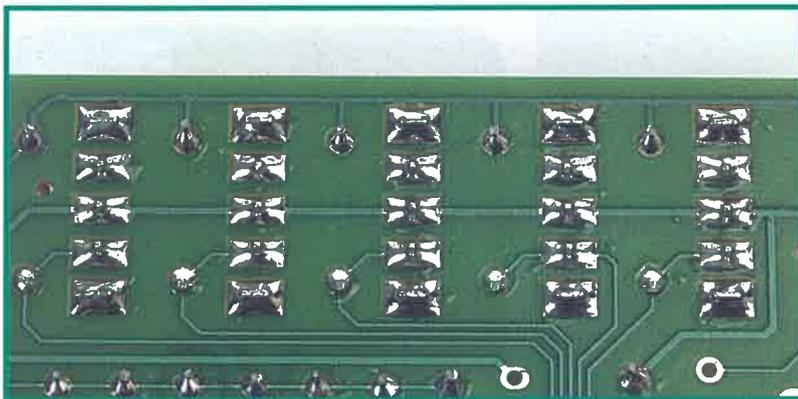
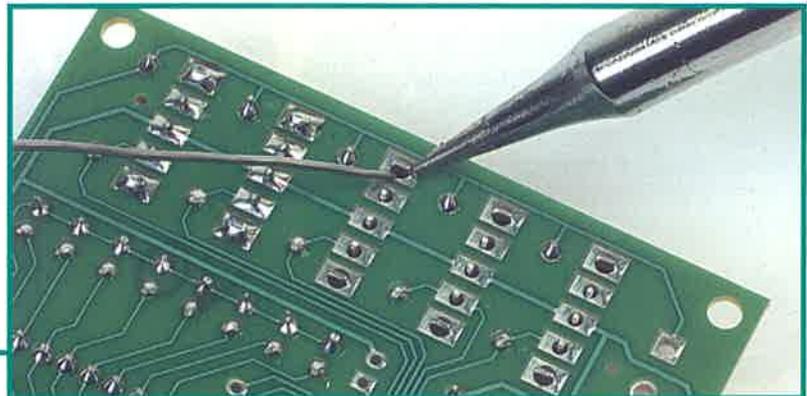
Dopo aver saldato le resistenze procederemo a saldare i commutatori. Non è importante il verso con cui li inseriremo, raccomandiamo solamente, per questioni di estetica, di montarli tutti con il testo impresso sul fianco, girato dalla medesima parte.





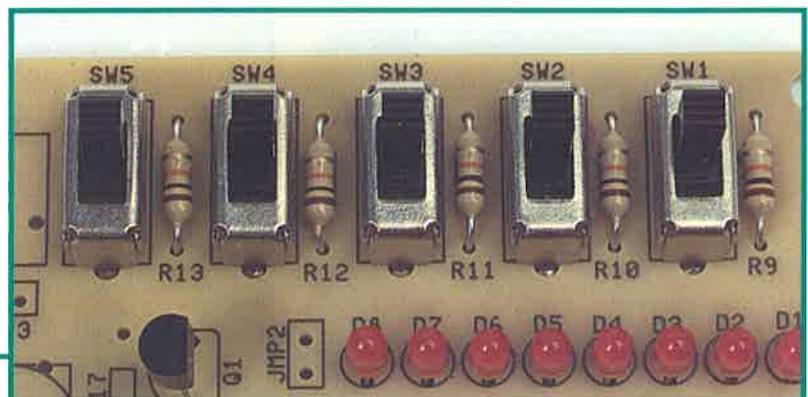
Collocheremo gli interruttori da SW2 a SW5. Dobbiamo premere sino a che risultino totalmente inseriti, e appoggino bene sulla superficie della scheda.

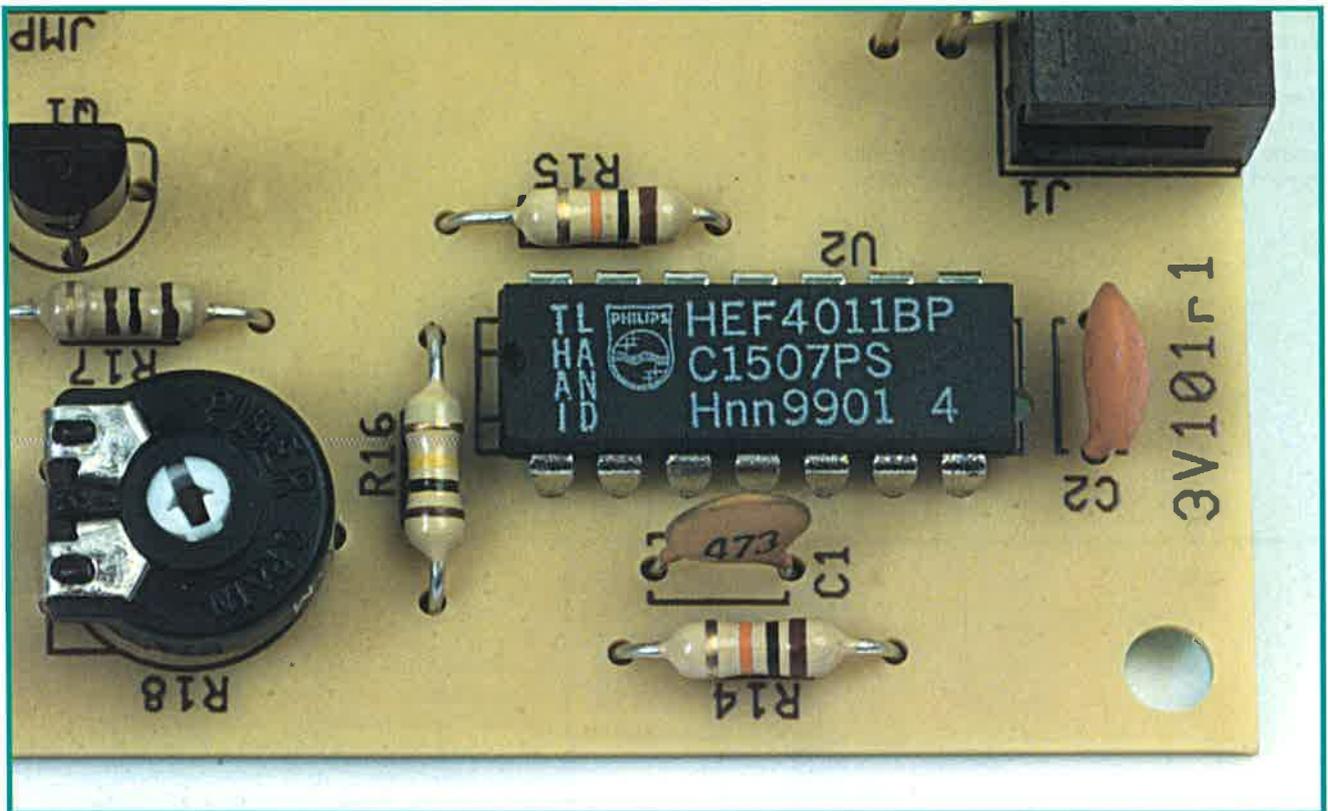
Dobbiamo saldare i commutatori sino a coprire con lo stagno le cinque piazzole di ognuno di essi. Di queste piazzole solo le tre centrali hanno una funzione nell'elettronica della scheda, le due esterne sono piazzole di appoggio, per dare maggiore solidità al posizionamento del commutatore. Per questo le salderemo tutte e cinque.



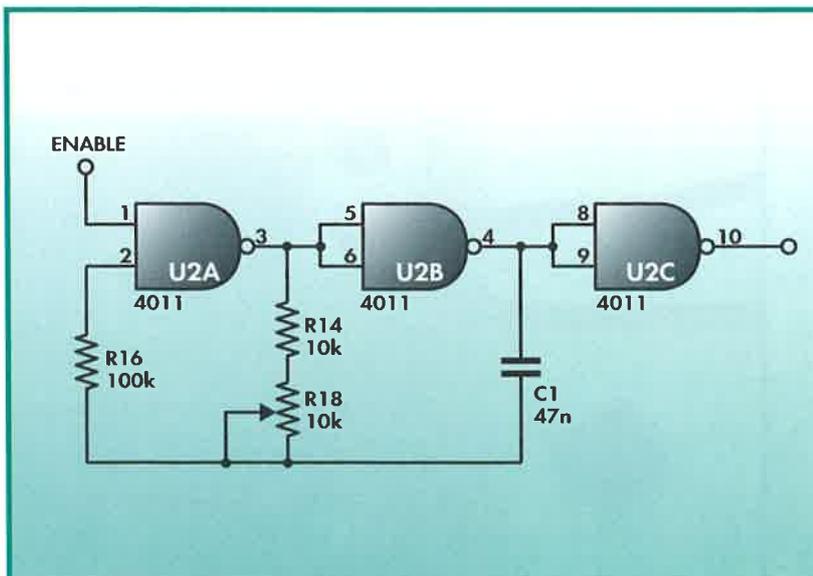
Questo è l'aspetto che devono presentare le saldature dei cinque interruttori, con le loro cinque resistenze da 10 K Ω associate. Dobbiamo assicurarci che le piazzole dei commutatori siano totalmente riempite, e non presentino soffiature.

I commutatori e le resistenze devono presentare questa disposizione. Le resistenze a filo della scheda, con la banda dorata orientata nel medesimo verso, i commutatori perpendicolari alla superficie della scheda, e solidamente saldati.



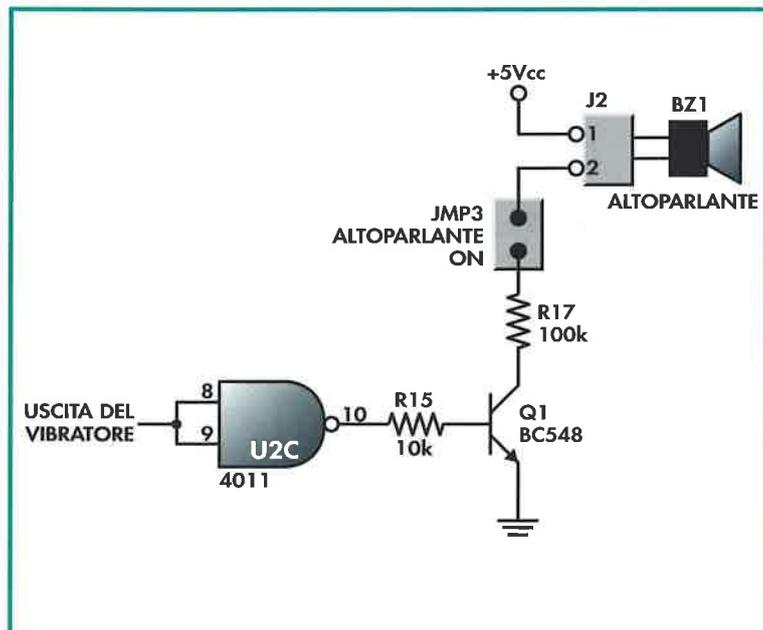
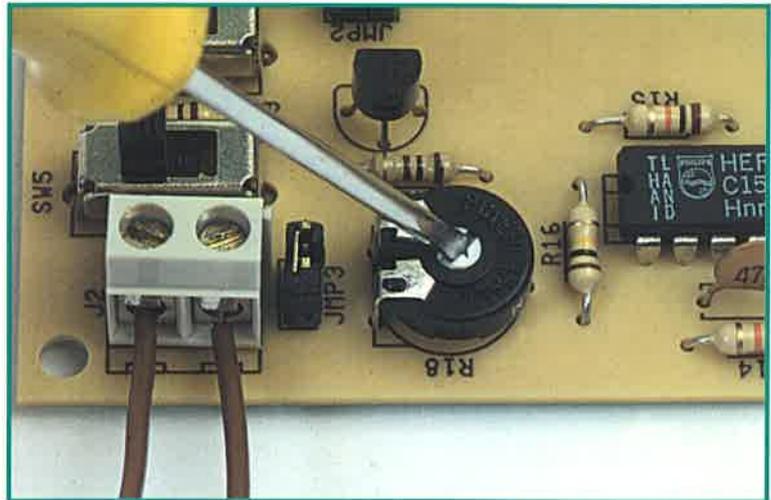


Per fare in modo che un altoparlante produca un suono, deve vibrare e per questo lo dobbiamo pilotare tramite un segnale elettrico. Possiamo programmare il PIC perché produca questo segnale, però porterebbe via molto tempo alla CPU, tempo che dobbiamo impiegare per altri lavori. Un altro problema è la capacità di carica del PIC: non è sufficiente per alimentare direttamente un altoparlante, quindi conviene considerare una soluzione hardware, che consiste nel far generare questo segnale da un circuito esterno al PIC. In questo modo l'altoparlante si attiverà e disattiverà con un livello logico, e il circuito avrà sufficiente capacità di corrente per alimentarlo.



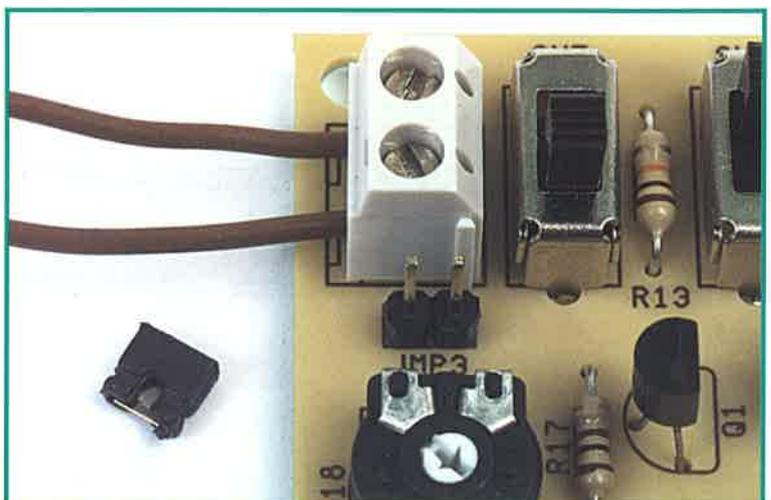
Questo è lo schema elettrico della parte del circuito vibratore che genera un'onda quadra. In seguito avremo un altro stadio che amplificherà questo segnale. Consta di diverse resistenze, un potenziometro, un condensatore e un circuito integrato 4011, che è una porta logica tipo NAND. Questa porta compie le funzioni della porta AND – già spiegata – però negate, quindi è come se collocassimo una porta NOT alla sua uscita. La filosofia del circuito consiste nell'utilizzare il tempo di carica e scarica del condensatore C1, per provocare un segnale oscillante '1' o '0'. Il tempo di scarica del condensatore dipenderà dal valore delle resistenze R14 e R18. Dobbiamo notare che se il segnale Enable è a '0', l'uscita di U2A sarà sempre a '1', di modo che non ci sarà condizione di oscillazione possibile e l'altoparlante non suonerà. La resistenza R18 è un potenziometro, per cui potremo modificarne il valore, producendo così un segnale periodico più o meno alto, il cui effetto sopra l'altoparlante sarà di ottenere suoni più o meno acuti.

Per modificare il valore del potenziometro raccomandiamo di usare un cacciavite a taglio, con cui potremo girarlo facilmente. Però bisogna fare attenzione, dato che il potenziometro non può dare un giro completo, perché ha un perno di fine giro in entrambe le direzioni.

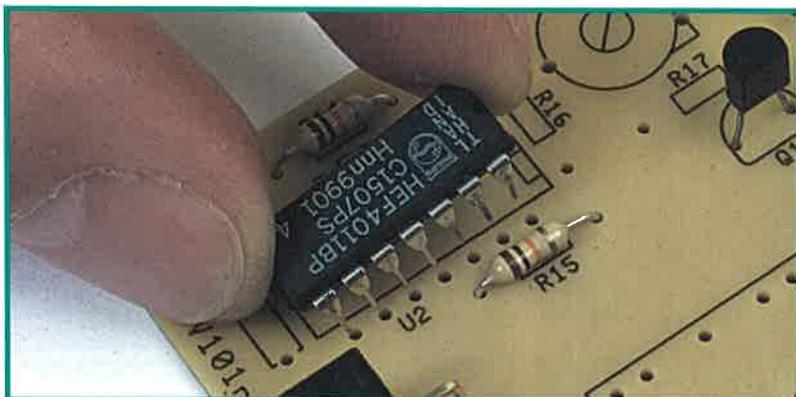


La funzione della porta U2C è semplicemente di ricostruzione, per ottenere un segnale perfettamente quadrato in uscita, che servirà per pilotare il transistor. Il transistor è di tipo NPN, quindi la sua base ha necessità di ricevere corrente. Per questo motivo, quando all'uscita di U2C avremo '1', il transistor si polarizzerà e lascerà passare corrente tra collettore ed emettitore, e come risultato alimenterà l'altoparlante. Grazie a questo stadio di amplificazione, stiamo applicando il segnale periodico ottenuto all'uscita del vibratore e sufficientemente amplificato all'altoparlante, perché possa muovere la sua membrana ed emettere suono.

Per fare in modo che l'altoparlante non emetta alcun suono, non è sufficiente fermare il funzionamento del circuito vibratore, dato che piccole correnti parassite che passano per il collettore del transistor, possono far sì che l'altoparlante emetta un leggero ronzio. Per questo motivo, quando vogliamo che l'altoparlante non suoni, possiamo togliere il jumper di JMP3, in questo modo scollegheremo l'altoparlante; isolandolo non emetterà più nessun suono. Grazie a questo jumper evitiamo di scollegare l'altoparlante dai suoi terminali quando non vogliamo utilizzarlo.

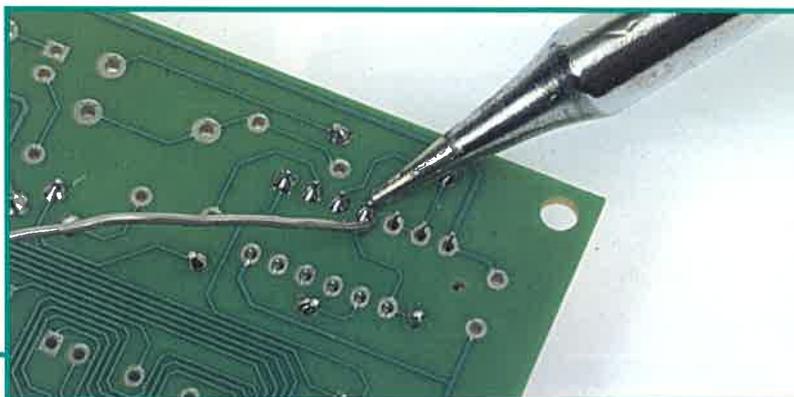


Montaggio passo a passo



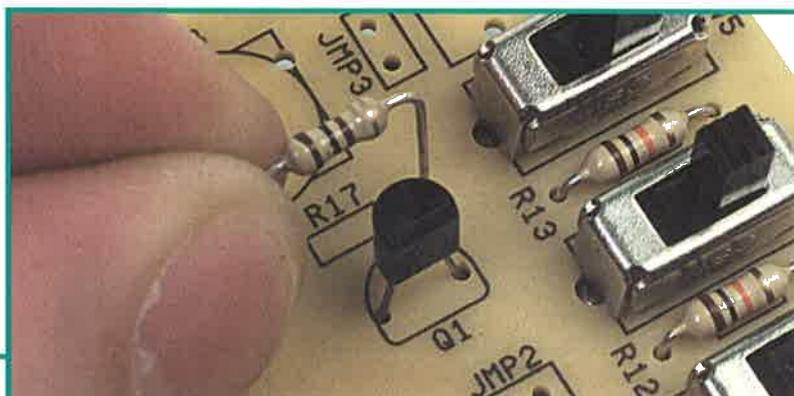
Per prima cosa inseriremo il circuito integrato 4011 in U2. Il verso d'inserzione del chip è segnalato sulla scheda da un quadrato collocato nella sua parte superiore. Quindi faremo coincidere la tacca di riferimento presente su uno dei lati del chip con il simbolo disegnato sulla scheda.

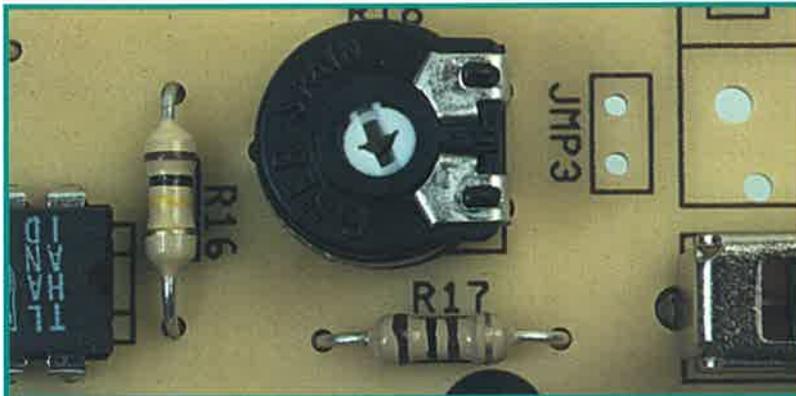
Per saldare il circuito integrato si raccomanda di iniziare dai due estremi opposti. Dopo aver realizzato queste saldature verificheremo che il verso di inserzione del chip sia corretto e che sia a filo della scheda; successivamente procederemo alla saldatura di tutti gli altri piedini.



Ora monteremo la resistenza da 100 K Ω che va inserita in R16. Il codice a colori di questa resistenza è marrone-nero-giallo. Come possiamo vedere, questo codice è molto simile alla resistenza da 10 K Ω , cambia solo l'ultimo colore (fattore di moltiplicazione) che è quello che fa la differenza fra i due valori. A volte, a causa della tonalità dei colori, è facile confonderli.

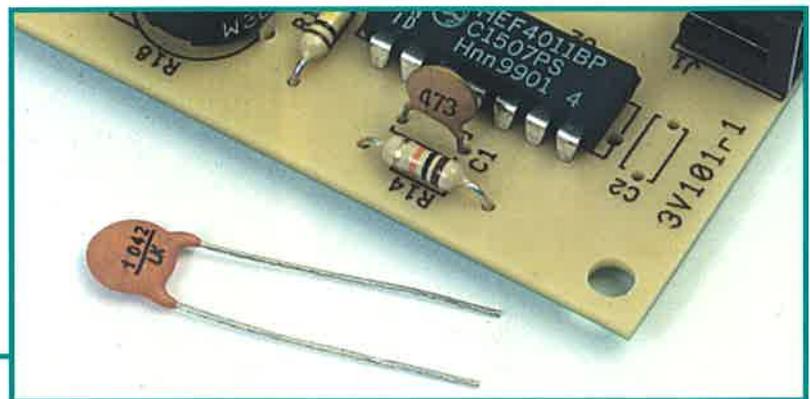
L'altra resistenza che dobbiamo saldare va montata su R17. Il suo valore è di 100 Ω , e il suo codice a colori è marrone-nero-marrone. Orientiamo la banda dorata nello stesso senso in cui abbiamo montato il resto delle resistenze sulla scheda.



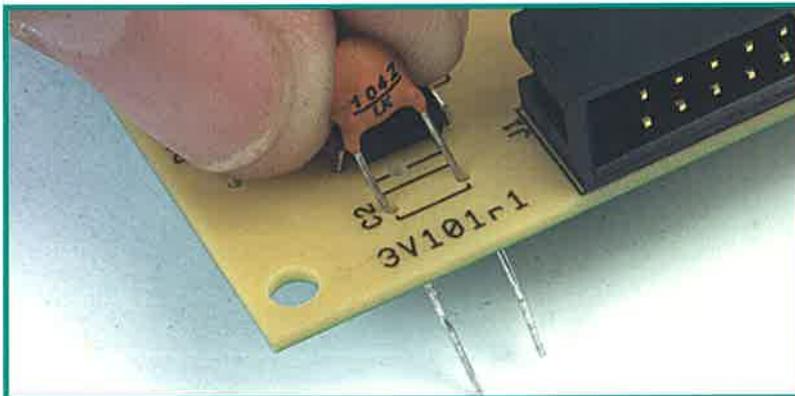


In seguito installeremo il potenziometro da 10 k Ω , che nella scheda è identificato come R18. Prima di saldarlo dobbiamo assicurarci che sia ben introdotto e non sia inclinato.

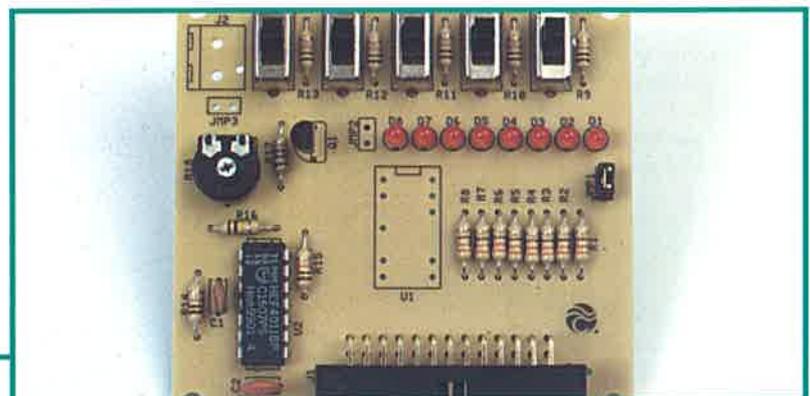
Saldiamo ora i condensatori. Prima salderemo quello da 47 nF, che verrà inserito in C1. È un condensatore ceramico che ha serigrafato il numero 473. Questo condensatore non ha polarità per cui è indifferente il verso con cui lo si inserisce.



Il prossimo condensatore che dovremo saldare va montato in C2 e il suo valore è di 100 nF. Ha serigrafato il numero 104Z, e come C1, è ceramico e non ha polarità.



Terminata questa serie di montaggi, avremo già saldato tutte le resistenze e i condensatori della scheda di ingresso e uscita. Inoltre avremo completato il montaggio del vibratore monostabile che utilizzeremo per il cicalino.



Di seguito spiegheremo il funzionamento del software di programmazione che serve per programmare il microcontroller. Per questo, abbiamo bisogno dei programmi compilati e senza errori mediante l'uso di un altro programma compilatore come il MPASM.

Il programma che dobbiamo eseguire per scrivere il PIC si chiama 16F82.EXE e si trova nelle directory PROGRAMMI del CD-ROM, così come mostrato dall'immagine. Raccomandiamo di copiare il programma in una cartella del disco rigido e di eseguirlo da lì, invece che dal CD-ROM. La directory radice per trovare il file del programma da scrivere sul PIC, sarà quella della directory in cui il programma si trova e dalla quale si eseguirà il programma 16F82.EXE, per cui è consigliabile posizionarlo nella stessa directory in cui si trovano i nostri file compilati, o in un'altra vicina. Questa è la finestra che appare dopo aver eseguito il programma. È divisa in varie zone

a partire dalle quali possiamo selezionare distinte funzioni di programmazione: il programma da scrivere, con le verifiche dello stato del PIC, ecc. In seguito spiegheremo in dettaglio ognuna di queste parti del software. Ricordiamo anche che questo software di programmazione, richiede che il microcontroller PIC sia inserito nella scheda di controllo, e che quest'ultima deve essere collegata alla porta LPT1 del computer, mediante un cavo DB25, e correttamente alimentata.

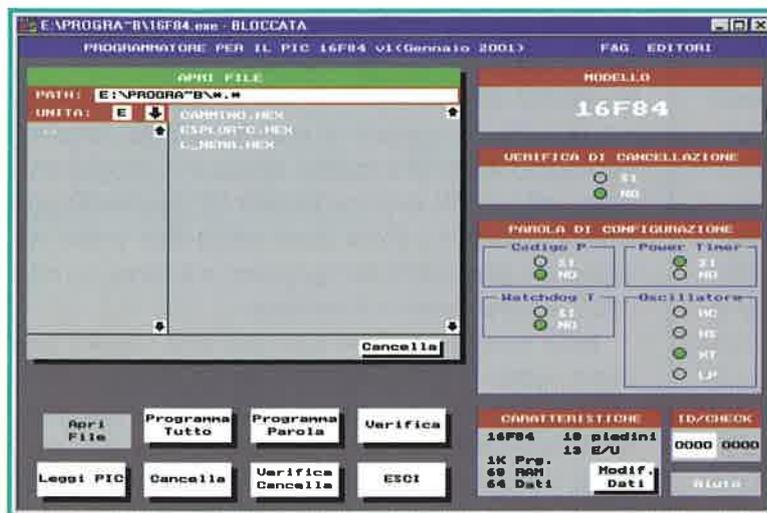
Nella finestra superiore a sinistra della videata, possiamo vedere il codice esadecimale del programma da scrivere. Questa è la sequenza dei dati che viene introdotta nel microcontroller, che si ottiene dopo aver compilato il programma. Quando eseguiamo il programma, tutte le posizioni sono riempite da un codice 3FFF di default. Questo corrisponde ad un codice di istruzione del PIC (14 bit per codice) composti da livelli '1' (memoria cancellata).



Posizione del programma all'interno del CD.



Programma in esecuzione.



Cercando il file.

Codice del programma.



Per aprire il file .hex che vogliamo scrivere nel PIC, dobbiamo cliccare sul pulsante apri, che si trova nella zona inferiore della finestra, dove ci sono tutti i pulsanti.

Dopo avere cliccato con il mouse questo pulsante, appare una finestra come quella dell'immagine. Come standard, ci troviamo ora nella directory in cui si trova il software di programmazione, e dalla quale lo abbiamo eseguito. Dobbiamo ora cercare l'ubicazione dei file .hex che vogliamo scrivere, muovendosi sull'albero delle directories del disco rigido. Vedremo sulla parte destra il nome dei files .hex che abbiamo nella directory e che potremo utilizzare, nel caso ne esista qualcuno.

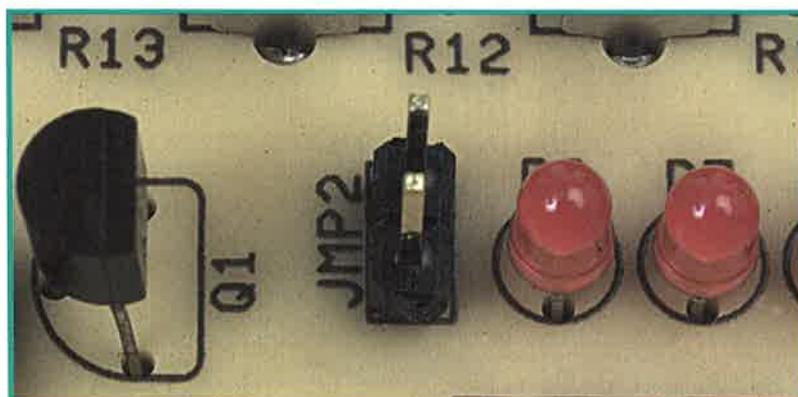
Per retrocedere nella directory, sarebbe a dire per avvicinarci alla directory base, dobbiamo cliccare con il mouse sopra i due punti che si trovano all'inizio, come prima directory, di ogni finestra. Per entrare in una

directory dobbiamo cliccare con il mouse sulla directory desiderata una volta che abbiamo visto il file .hex che vogliamo scrivere; cliccando su di esso lo manderemo direttamente al buffer del software di lettura per mandarlo nel PIC.

Dopo aver aperto un file .hex, nella finestra che rappresenta il buffer di memoria, nella posizione dove prima avevamo solo dati 3FFF, devono apparire una serie di numeri esadecimali che corrispondono al codice hex che scriveremo nel PIC.

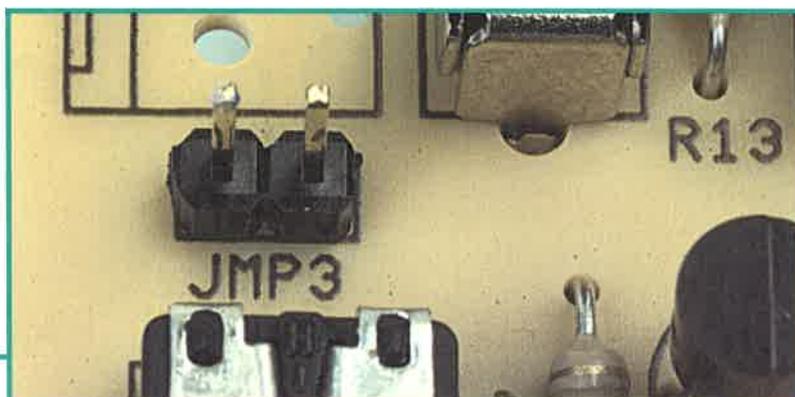
Possiamo vedere tutto il codice del programma, che arriva sino all'indirizzo 3FF. Per questo possiamo cliccare sui pulsanti Pag su o Pag giù o sopra le due frecce. Con le frecce scenderemo o saliremo solo di una linea per volta. Con i pulsanti Pag su o Pag giù, saliremo o scenderemo di una videata, in modo da poterci muovere sul file molto più rapidamente.

Montaggio passo a passo

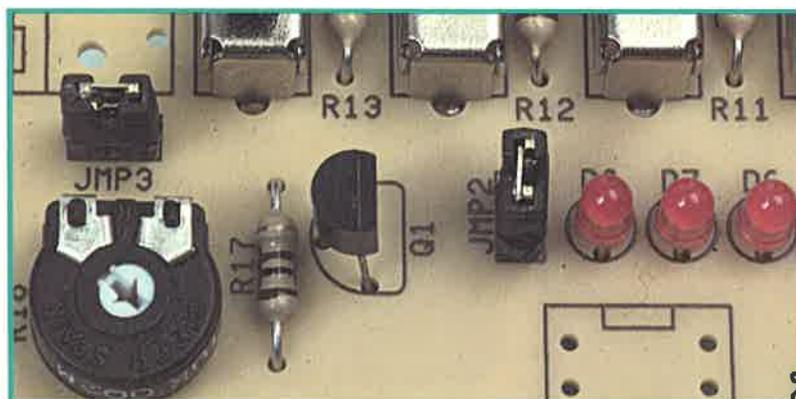


Come prima cosa monteremo il connettore a due vie in JMP2. Dobbiamo introdurlo sulla scheda dalla parte corta dei pin. Dobbiamo saldarlo in maniera che rimanga perpendicolare rispetto alla scheda.

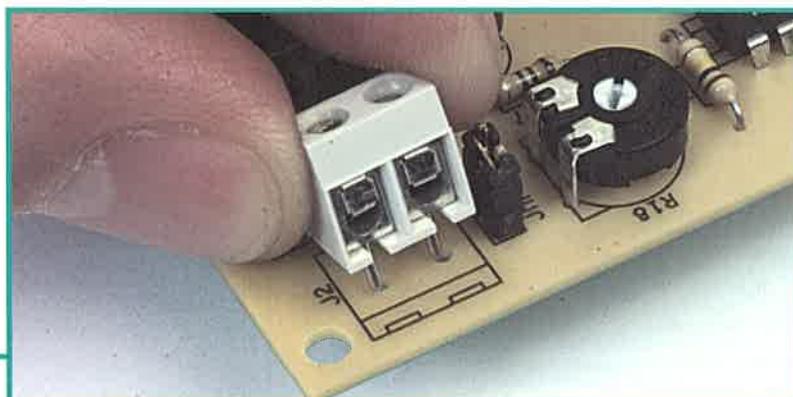
Ripeteremo il processo con l'altro connettore a due vie, che va collocato in JMP3. Raccomandiamo di saldare solo un pin all'inizio, per verificare che rimanga perpendicolare, e dopo saldare entrambi i piedini per finire il fissaggio.



Ora collegheremo i due jumpers sopra i due connettori che abbiamo saldato in JMP2 e JMP3. Al momento non c'è bisogno di chiudere questi circuiti, però lasciandoli montati sopra i jumpers, eviteremo di perderli.



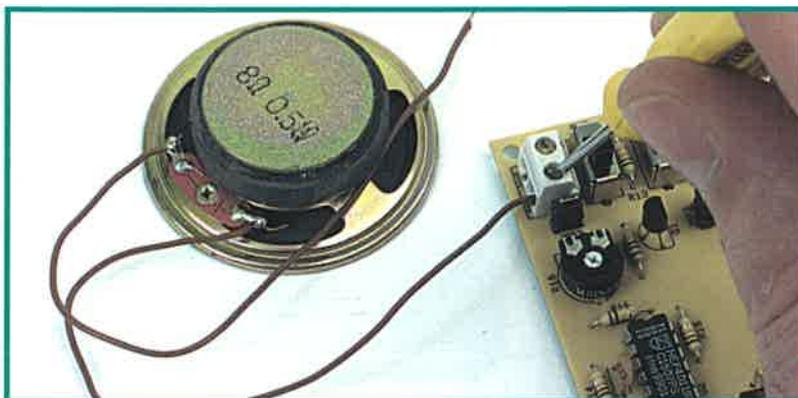
La morsettiera a due contatti servirà per collegare gli altoparlanti alla scheda. Dobbiamo collocarla in J2, secondo il verso indicato nell'immagine, cioè con i fori per l'ingresso dei cavi verso l'esterno della scheda.





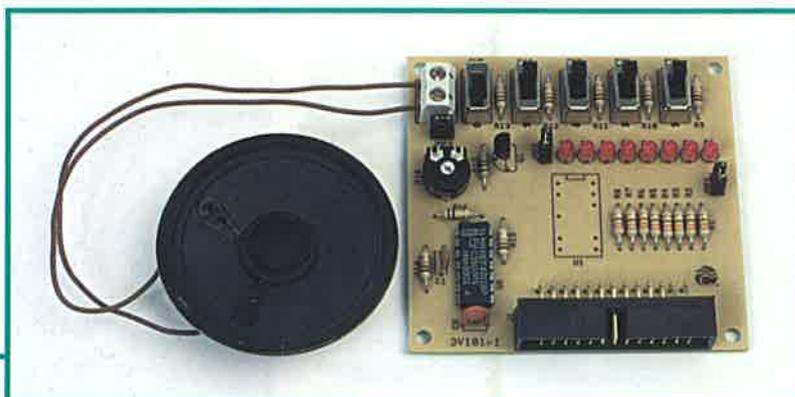
Le saldature della morsetteria hanno bisogno di più stagno delle saldature normali dei diodi e delle resistenze. Dopo aver realizzato la saldatura non è necessario tagliare il resto dei piedini.

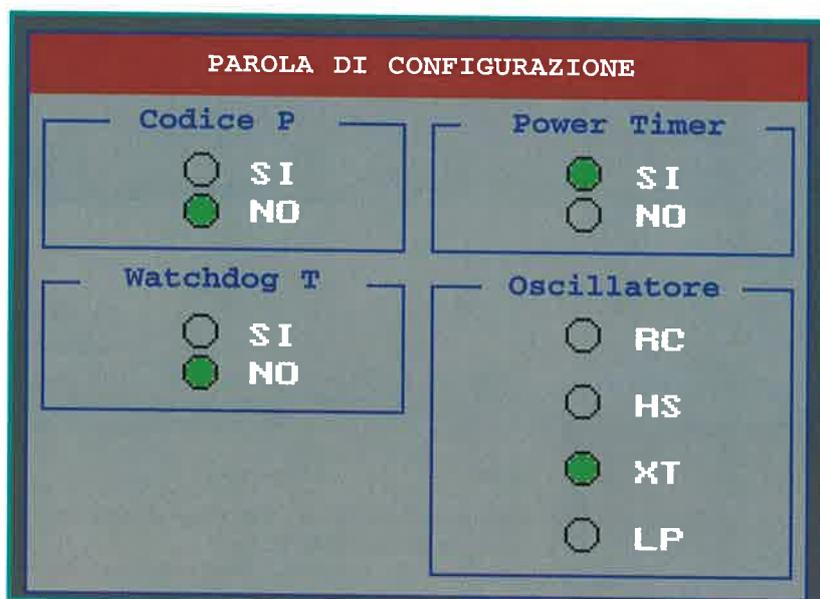
Dovremo saldare ora i cavi dell'altoparlante. Raccomandiamo di stagnare sia i cavi sia la superficie dell'altoparlante dove realizzeremo la saldatura (prestagnatura). Ora decideremo il cavo da saldare, riscaldere la superficie, e aspetteremo che si unisca all'altoparlante.



L'altoparlante ha un terminale positivo e l'altro negativo. Il positivo ha un segno con il simbolo +. Questo terminale positivo deve essere montato nell'ingresso della morsetteria indicato dall'immagine, quello della parte situata più vicino a JMP3.

Una volta montato il terminale positivo, monteremo anche quello negativo. La scheda di I/O è così completata; manca solamente il montaggio del display a sette segmenti, come si vede nell'immagine.





Mediante questo menù possiamo selezionare la parola di configurazione che potremo memorizzare nel microcontroller. La prima opzione, codice P, serve per proteggere o non il codice di programma che vogliamo scrivere. Se attiviamo la protezione del codice e scriviamo il PIC, quando lo leggeremo appariranno tutti gli indirizzi con il dato 0000, sarebbe a dire il codice è criptato. Il Watchdog T è il temporizzatore interno che può essere utilizzato per evitare blocchi nel PIC, e il Power Timer consiste nel comandare al PIC di realizzare una piccola temporizzazione dopo il Power On, in modo che la tensione ai suoi ingressi si stabilizzino prima di iniziare ad eseguire il programma. Infine dobbiamo selezionare il tipo di oscillatore con il quale lavorerà il microcontroller, nel nostro caso sceglieremo sempre l'oscillatore XT.



Nella parte inferiore della finestra corrispondente al buffer di memoria troviamo i pulsanti attraverso i quali realizziamo tutte le funzioni di programmazione. Inizieremo la loro descrizione dalla fila superiore.

Il primo pulsante "Apri File" serve per caricare il programma esadecimale (estensione .hex) che vogliamo memorizzare nel microcontroller PIC. Il secondo pulsante, "Programma Tutto", ci permette

di scrivere nel PIC sia la memoria di programma con il file .hex selezionato, che la parola di configurazione. Il pulsante "Programma Parola" scrive solo la parola di configurazione del PIC, senza modificare il contenuto della memoria Flash di programma.

L'ultimo pulsante di questa fila, "Verifica" serve per verificare se abbiamo scritto bene

il PIC e se è stato o no correttamente programmato. Si confronta il contenuto del buffer con quello della memoria interna.

Il pulsante della seconda fila "Leggi PIC", ci mostra nella finestra

del buffer il contenuto di un PIC già programmato. Leggeremo sia la memoria di programma del PIC che la sua parola di

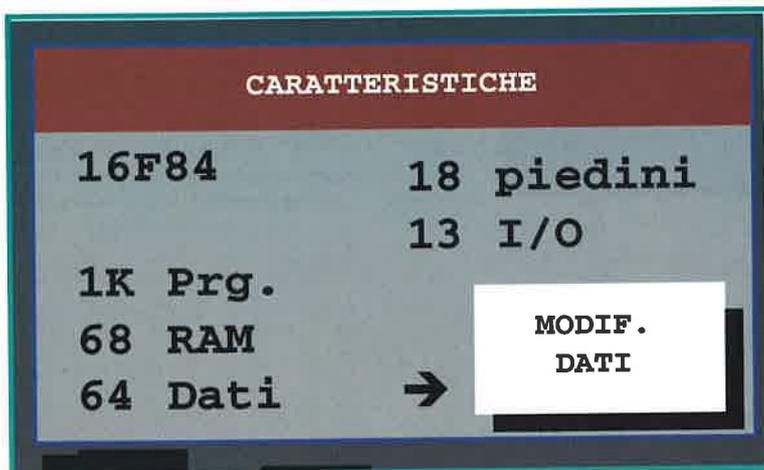
configurazione. Il pulsante "Cancella" pulisce la memoria di programma del microcontroller ed elimina la parola di configurazione.

Mediante il pulsante "Verifica Cancella" verifichiamo se la memoria del PIC

è cancellata o meno. Infine mediante il pulsante "Esci" terminiamo l'esecuzione del software di programmazione.

Consigli pratici

Se attiviamo l'opzione "Verifica Cancella" il software si assicura che il PIC sia cancellato prima di procedere alla scrittura di ognuno degli indirizzi di memoria di programma. Questa verifica si deve fare sempre manualmente mediante il pulsante, ed è molto affidabile.



Nella parte inferiore del programma vediamo un riassunto delle caratteristiche più importanti del microcontroller PIC16F84, per il quale è stato realizzato un software di programmazione. Così possiamo vedere che possiede 1 K di memoria flash di programma, 64 byte per i dati EEPROM, 13 linee di I/O ecc.

A lato delle caratteristiche appare un campo con nome ID/CHECK. Si tratta di due numeri esadecimali da quattro bit che si scrivono in una zona riservata della memoria del PIC.

Nel numero ID si introduce un numero libero, che generalmente è utilizzato per indicare il numero di versione del programma scritto, identifica il fabbricante, ecc. Il checksum ci può servire come codice di sicurezza, per verificare se la memoria del PIC ha subito alterazioni, o se si trova memorizzata così come è stata scritta.

Quando scriviamo il PIC, il programma calcola mediante un algoritmo, il codice checksum, in funzione dei dati che sono scritti nella memoria.

Quando leggiamo la memoria del PIC, si calcola impiegando lo stesso algoritmo, il checksum corrispondente ai dati letti che sono memorizzati. In questo modo, se il numero che mostra la finestra dopo aver letto il PIC è lo stesso che abbiamo visto quando abbiamo scritto il programma, significherà che la memoria del PIC è rimasta inalterata.



MODULO DI INGRESSO-USCITA

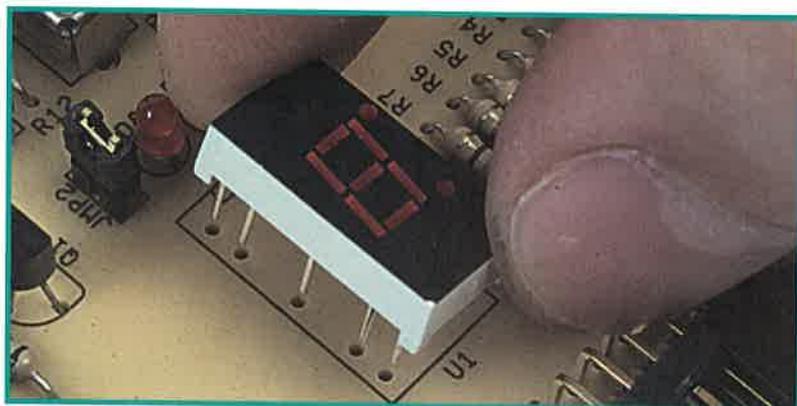
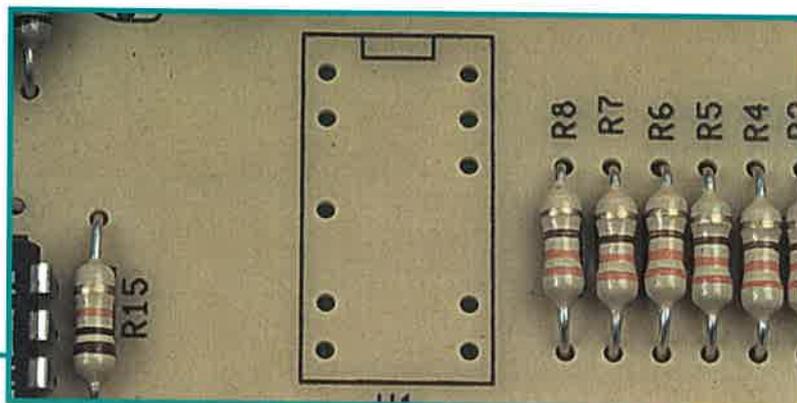
PL 38

Montaggio passo a passo



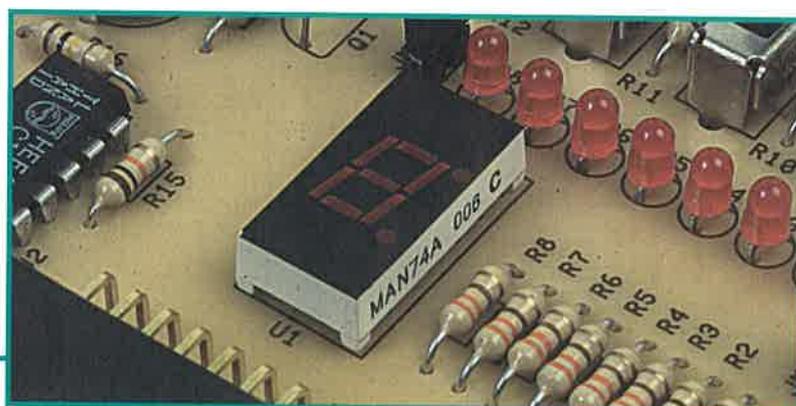
Montiamo ora l'ultimo componente che ci resta da mettere sulla scheda di Ingressi e Uscite: il display a sette segmenti. Si tratta di un display a codo comune.

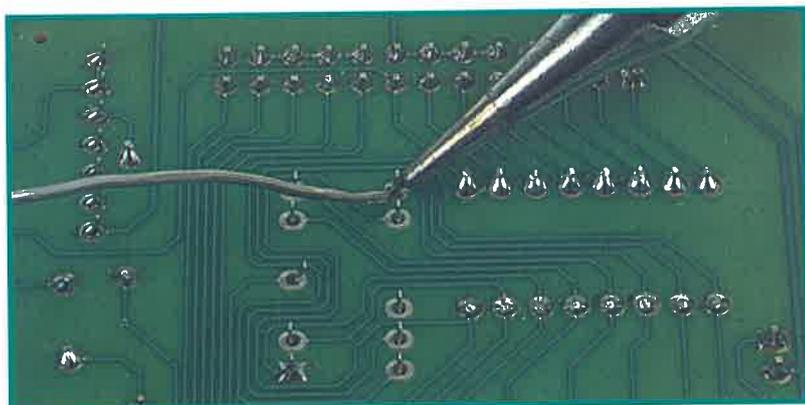
Dobbiamo inserire il display in U1. Possiamo osservare che si tratta di uno spazio che non ha una foratura regolare, dato che ha solo i fori che coincidono con la distribuzione dei piedini del display.



Dobbiamo inserire il display in modo che i piedini di questo coincidano con i fori di U1. Questo sarà il verso corretto e l'unico modo di inserimento possibile.

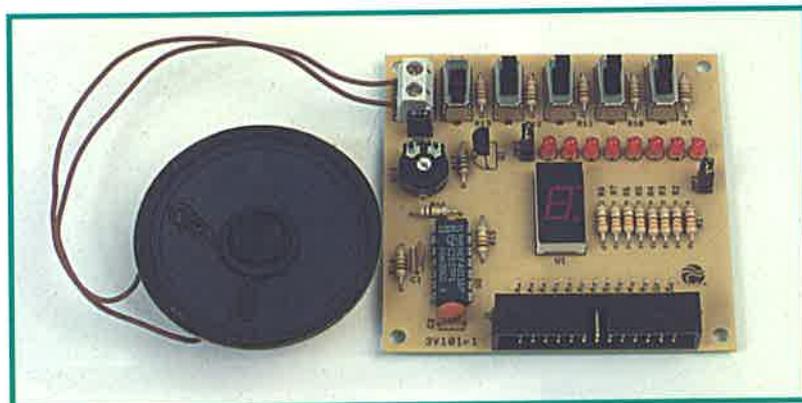
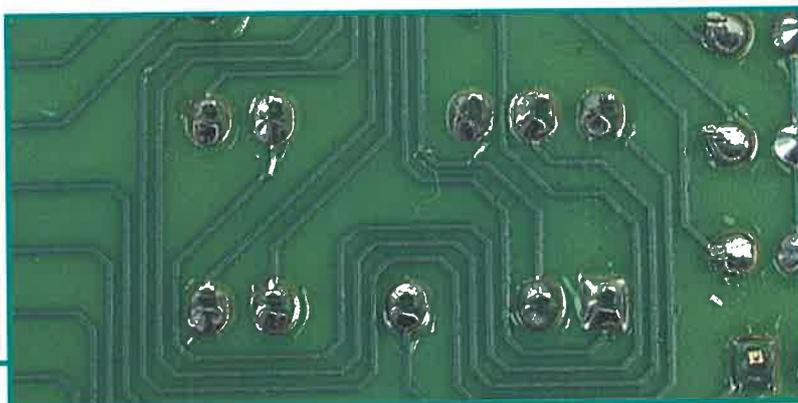
Introdurremo il display fino a che appoggi completamente sulla scheda. In realtà dispone di quattro punti di contatto ai lati, i quali devono risultare ben appoggiati sulla superficie.





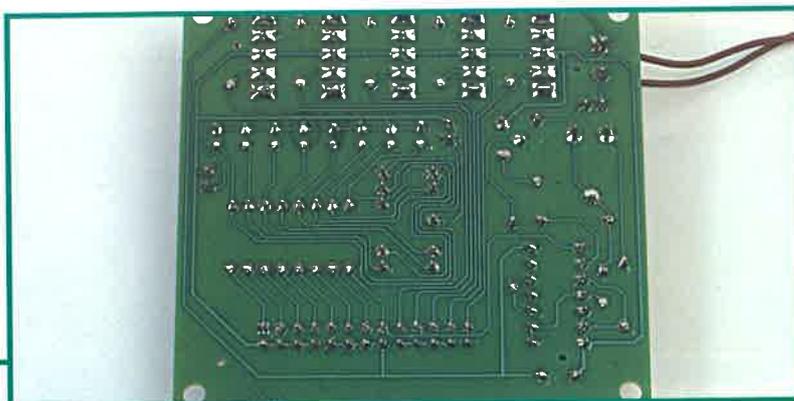
Prima salderemo solo i due angoli del display. Una volta realizzata questa saldatura, verificheremo che il display sia inserito completamente nella scheda e non sia rimasto storto.

Una volta verificato il corretto posizionamento del display procederemo a realizzare il resto delle saldature, che dovranno essere brillanti e a forma di cono, come mostrato nell'immagine.



Abbiamo terminato il montaggio della scheda di Ingressi e Uscite. Questa ci servirà per imparare a programmare il microcontroller PIC e per provare la validità dei nostri programmi. Dobbiamo avere tutti i componenti saldati e distribuiti come mostrato nell'immagine.

Questo è l'aspetto che devono avere le saldature della scheda di Ingressi e Uscite. Dobbiamo verificare che siano brillanti, proporzionate e che non ci siano contatti fra di loro, per assicurarci del buon funzionamento della scheda prima di iniziare a provare i programmi.

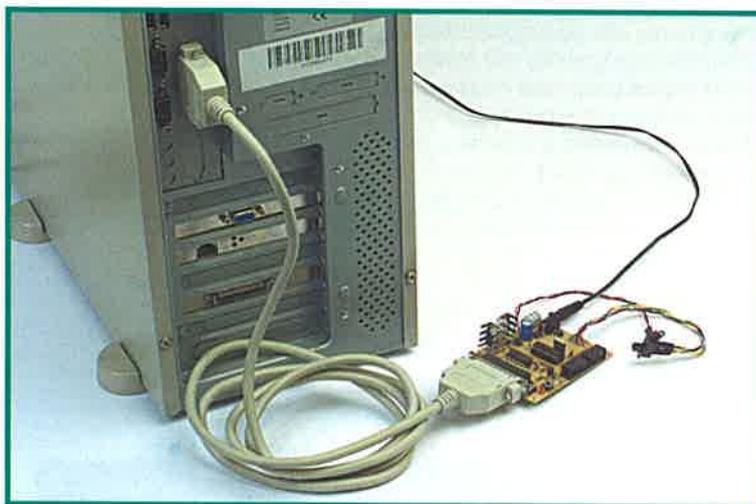




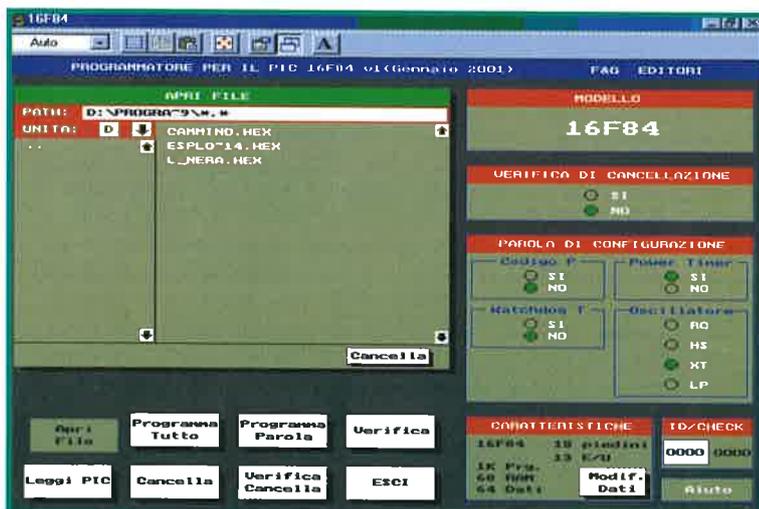
Con il programma 16F84.EXE oltre a scrivere la memoria Flash e la parola di configurazione del PIC possiamo anche vedere il contenuto della memoria EEPROM dei dati. Il PIC consta di 64 indirizzi di memoria EEPROM, nella quale possiamo scrivere, mediante il nostro programma di applicativo, valori che non vogliamo che vadano persi quando al microcontroller viene tolta tensione. Per accedere alla videata mostrata nell'immagine, dobbiamo cliccare il pulsante "Modif. Dati", che si trova all'interno del menù "Caratteristiche". Dopo aver premuto questo pulsante si apre un'altra finestra in cui sono spariti alcuni pulsanti nella parte inferiore, inoltre si riducono anche gli indirizzi di memoria che possiamo leggere. In concreto possiamo leggere i 64 byte che abbiamo memorizzato negli indirizzi EEPROM del PIC.

PASSI DA SEGUIRE PER PROGRAMMARE IL PIC

Il primo passo consiste nell'assicurarci che il microcontroller PIC sia inserito nella scheda di controllo, e che sia orientato nella posizione corretta. Ora dobbiamo collegare la scheda al PC mediante un cavo con due connettori DB25 (uno maschio e uno femmina), come mostrato nell'immagine. Questo cavo deve essere acquistato a parte in un negozio specializzato. Dobbiamo anche assicurarci che il connettore SW3 della scheda di controllo si trovi nella posizione di scrittura, altrimenti il programma ci darà messaggi tipo "Errore di verifica" oppure "Il PIC non è cancellato" quando si tenterà di programmare il PIC. Dopo aver realizzato questi tentativi, alimenteremo la scheda e posizioneremo gli interruttori su On. Per il processo di memorizzazione raccomandiamo di utilizzare un trasformatore da 12 V, o di montare tutte le batterie nell'apposito vano sul telaio. Quando la scheda sarà alimentata e collegata al PC, apriremo il programma di scrittura. Dopo aver aperto il programma



dovremo selezionare un file .hex che si possa programmare nel PIC. Il microcontroller supporta sino a 1.000 cicli di programmazione e di cancellatura. Se avremo assemblato un programma senza errori con MPASM, disporremo di un file .hex che potremo scrivere nel PIC, nel caso non lo avessimo potremo selezionare, a titolo di prova, i file che si trovano nella directory PROGRAMMI del CD.



Questa è la finestra che si vede dopo l'accesso ai file della directory. Nella parte destra, sono mostrati i file .hex che si possono scrivere nel PIC. Ne selezioneremo uno qualunque, ad esempio ESPLORATORE.HEX. Prima di programmare bisogna cancellare il microcontroller e verificare la cancellazione. Per questo cliccheremo il pulsante "Cancella". Se tutto va bene, dovrebbe lampeggiare il diodo LED D9 della scheda di controllo, e pervenirci il seguente messaggio: "La cancellazione è stata realizzata". Adesso possiamo verificare se il PIC è stato cancellato cliccando il pulsante "Verifica Cancella"; nel caso la verifica sia negativa ripeteremo l'operazione di cancellazione. Non dovremo scrivere sino a che il PIC non sarà stato cancellato correttamente. Dopo ogni apparizione di un messaggio nel programma dovremo cliccare il pulsante "Continua" per proseguire con le operazioni successive.

LA CANCELLAZIONE È STATA REALIZZATA**3****CONTINUA***Ciclo di cancellazione realizzato.***IL PIC È CANCELLATO****4****CONTINUA***Verifica di cancellazione del PIC.***IL PIC NON È CANCELLATO****5****CONTINUA**

Quando il PIC sarà correttamente cancellato procederemo alla sua programmazione. Di seguito scriveremo il programma che abbiamo selezionato e che stiamo vedendo nella finestra del Buffer di Memoria di programma. Prima di scrivere il PIC ci dobbiamo assicurare anche che la parola di configurazione contenga le opzioni che vogliamo assegnare al nostro microcontroller. Ricordiamo che l'oscillatore deve sempre essere XT, cliccheremo il pulsante "Programma Tutto" per scrivere il programma. Durante il processo di scrittura del PIC il diodo D9 lampeggerà come quando lo stavamo cancellando. In funzione della lunghezza del programma, il tempo impiegato per la scrittura sarà maggiore o minore.

IL PIC È STATO PROGRAMMATO**6****CONTINUA**

Se il PIC si programma correttamente, apparirà il messaggio "Il PIC è stato programmato". Al contrario, se non si potrà programmare, apparirà un messaggio: "Errore di Verifica". Dopo aver scritto il PIC potremo verificare il contenuto del medesimo. Se appare il messaggio "La verifica è stata eseguita correttamente", il PIC sarà perfettamente programmato, e potremo eseguire il suo programma interno.

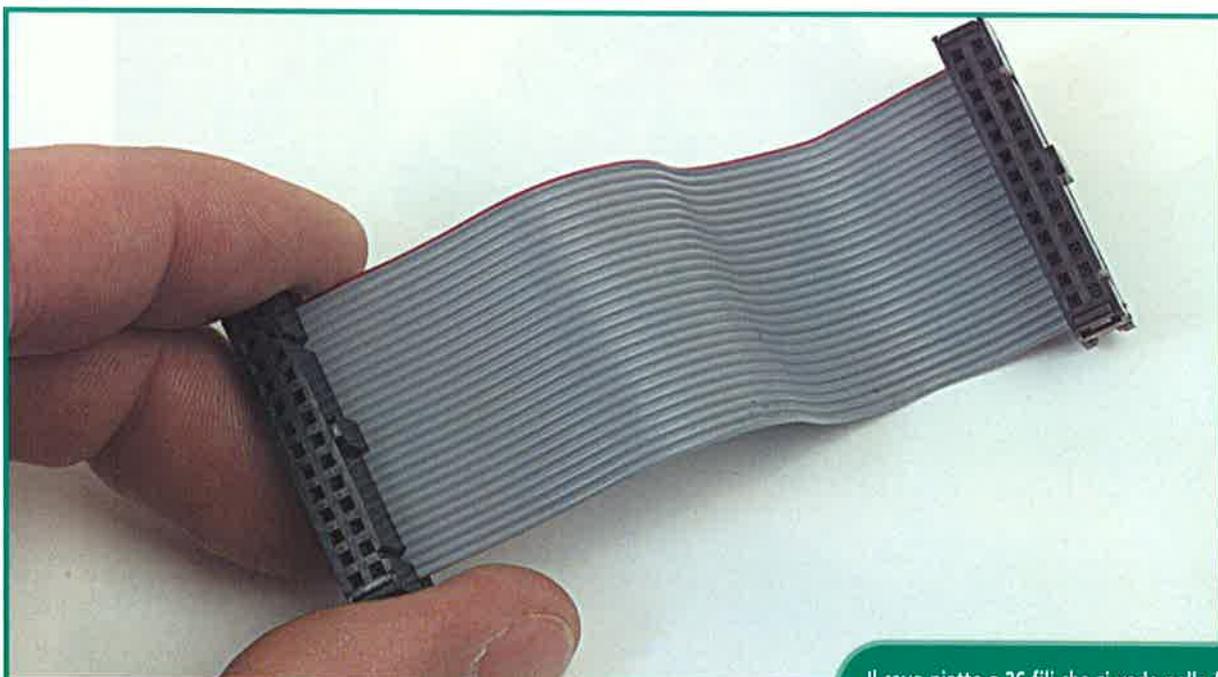
LA VERIFICA È STATA ESEGUITA CORRETTAMENTE**7****CONTINUA***La verifica ha avuto successo.***ERRORE DI VERIFICA****8****CONTINUA**

Dopo aver programmato il PIC, potremo modificare il contenuto della parola di configurazione, indipendentemente dalla memoria di programma. Per fare questo cliccheremo il pulsante "Programma Parola" e introdurremo nel PIC la nuova configurazione. Per eseguire il programma nella nostra applicazione, dobbiamo scollegare il cavo DB-25 dalla scheda di controllo (non è obbligatorio). Inoltre dobbiamo muovere la posizione SW3 da PROG a RUN. In questo modo le linee RB6 e RB7 rimangono pronte per essere collegate ai sensori e agli attuatori, invece che al circuito di scrittura della scheda di controllo. Ora potremo modificare la fonte di alimentazione e passare all'alimentazione mediante un trasformatore, o alimentare con pile o batterie. Il consumo in questo momento è minore e se avremo scollegato il cavo DB25, disporremo di un sistema autonomo. Questo è il processo che bisogna sempre seguire per la programmazione corretta del PIC, e dobbiamo seguirlo anche ogni volta che introdurremo un programma nuovo. Se abbiamo seguito questi passi e nonostante tutti i tentativi fatti non riusciamo a cancellare o programmare il PIC, dobbiamo rivedere le saldature e la disposizione dei componenti sulla scheda. Possiamo toccare con la punta del saldatore le saldature per verificare se qualcuna di esse non sia stata ben fatta o non abbia preso la forma di cono.

MODULO DI INGRESSO-USCITA

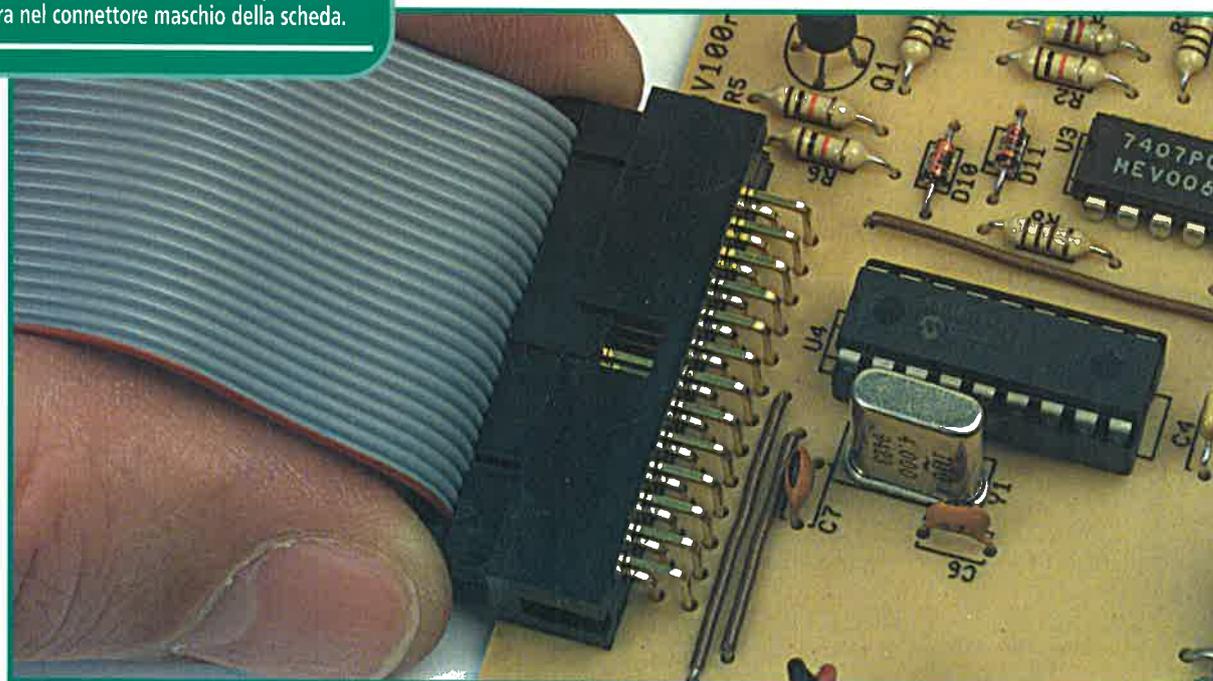
PL 40

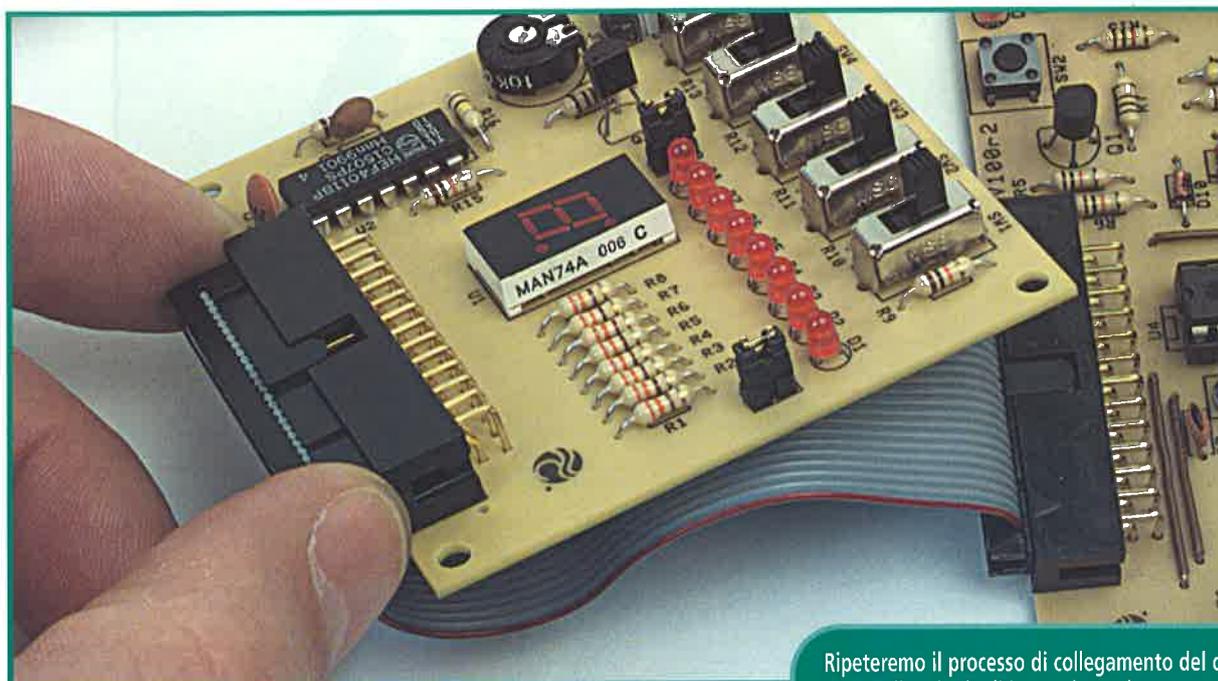
Montaggio passo a passo



Il cavo piatto a 26 fili che si vede nella figura, è il mezzo che permetterà al PIC di comunicare, tramite la scheda di controllo, con il resto delle schede di cui sarà costituito Monty. Attraverso questo cavo inoltre si trasportano le tensioni di alimentazione delle schede.

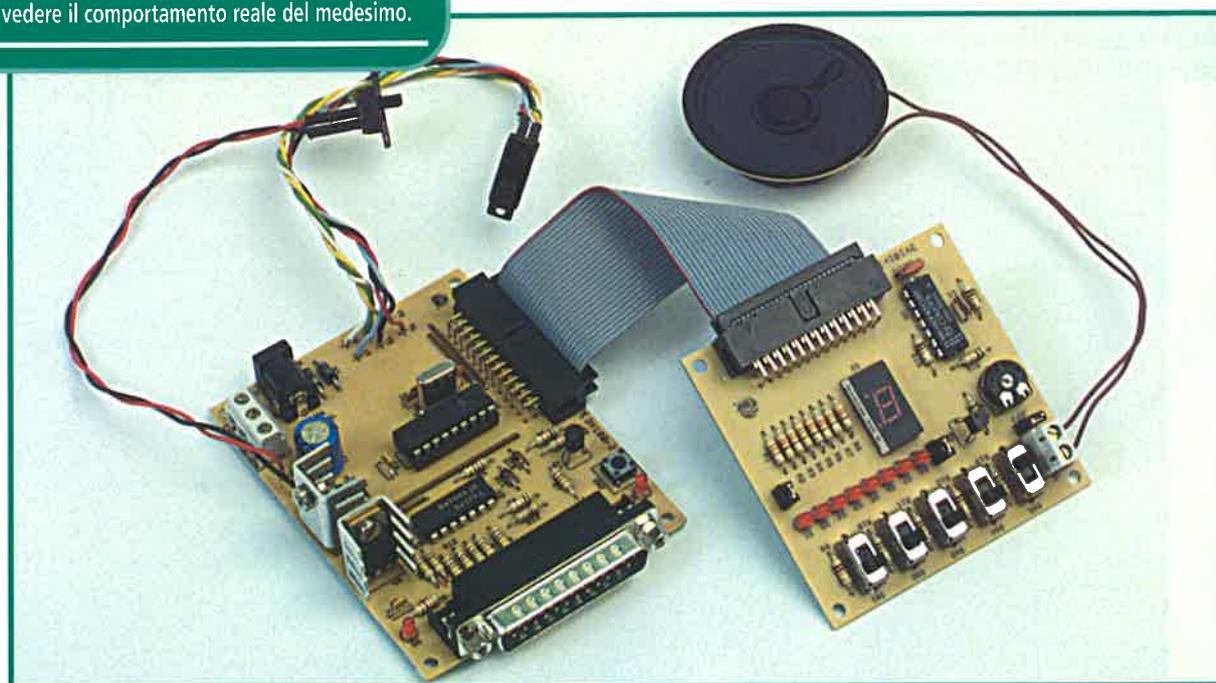
Dobbiamo collegare un estremo del cavo parallelo alla scheda di controllo e l'altro alla scheda di ingressi e uscite. Dobbiamo introdurre il cavo con la linguetta rivolta verso l'alto, nell'unica posizione in cui entra nel connettore maschio della scheda.

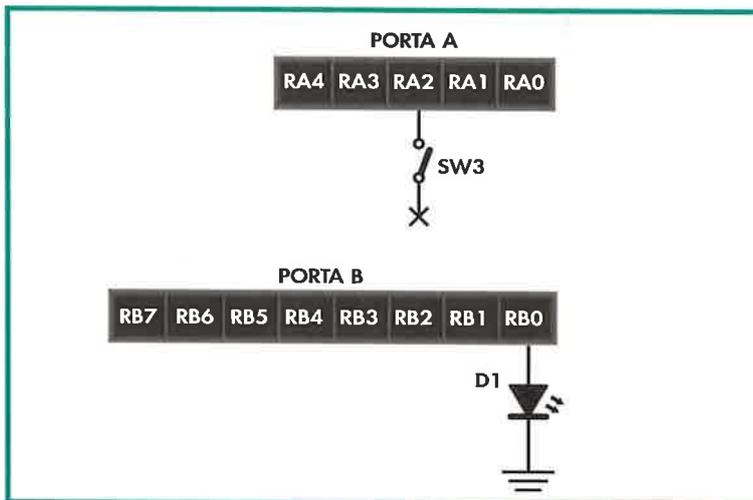




Ripeteremo il processo di collegamento del cavo piatto alla scheda di ingressi e uscite. Dobbiamo fare pressione fino a che il cavo sarà introdotto totalmente e in modo uniforme.

A partire da questo momento avremo collegato il microcontroller alla scheda di ingressi e uscite attraverso il cavo piatto, che denomineremo cavo PIC-BUS. In questo modo potremo verificare sulla scheda di ingressi e uscite la validità dei programmi che scriveremo nel microcontroller, per vedere il comportamento reale del medesimo.





Realizziamo ora il primo programma con gli interruttori presenti sulla scheda di ingressi e uscite. È un semplice esercizio di utilizzo degli ingressi e delle uscite. Consiste nel rilevare lo stato dell'interruttore SW3, e in funzione del medesimo accendere o spegnere il LED D1. Nella fotografia è mostrato lo schema utilizzato per la realizzazione dell'esercizio, che eseguiremo collegando, mediante il cavo PicBus, la scheda di controllo e quella di ingressi e uscite. Nello schema possiamo anche vedere la circuiteria minima necessaria per il funzionamento del PIC. Se ad un certo punto avremo bisogno di far nuovamente iniziare l'esecuzione del programma, non dovremo far altro che premere il pulsante di Reset.

Questa è l'intestazione del programma. All'inizio di ogni programma bisogna sempre definire il tipo di processore che vogliamo utilizzare, nel nostro caso il PIC16F84. Poi assegniamo le etichette agli indirizzi di memoria dei registri dei dati, per facilitare il loro successivo impiego durante la realizzazione del programma. Poniamo nel Vector 0 un'istruzione di salto all'inizio del programma. Con ORG 5, saltiamo il Vector di interrupt posizionato all'indirizzo 4, iniziando il programma all'indirizzo 5.

```

c:\progra~1\mplab\esempio1.asm
1 ;*****
2 ;Programma che accende o spegne il diodo
3 ;led D1 della scheda di ingressi e uscite
4 ;in funzione dello stato del commutatore SW3.
5
6     list    p=16F84        ;Tipo di dispositivo
7
8 Zona delle etichette
9
10 STATUS EQU    0X03
11 PORTA  EQU    0X05
12 PORTB  EQU    0X06
13 RP0    EQU    0X05
14
15
16     ORG    0                ;Vector Reset
17     goto  INIZIO
18
19     ORG    5
20

```

```

c:\progra~1\mplab\esempio1.asm
21
22
23 ;Inizio del programma
24
25 INIZIO: bsf     STATUS, RP0
26         clrf   PORTB
27         movlw  b'00000100'
28         movwf  PORTA
29         bcf   STATUS, RP0
30         clrf   PORTB
31
32

```

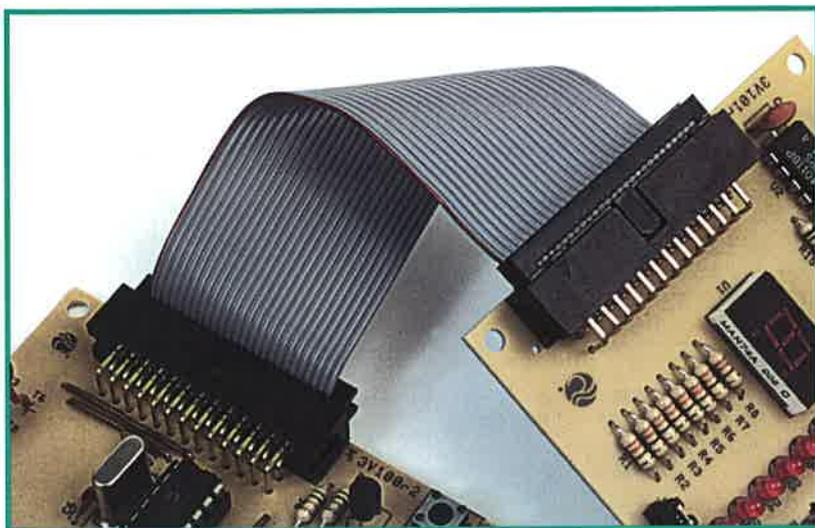
La prima cosa che dobbiamo fare all'inizio del programma è configurare i registri che vogliamo utilizzare. In questo caso dobbiamo configurare la porta B come uscita (il diodo D1 è collegato a RB0), e il pin RA2 come ingresso (pin al quale arriva il segnale dell'interruttore SW3). Per realizzare la configurazione delle porte passiamo al Banco 1, e dopo averle configurate torniamo al Banco 0 per continuare ad eseguire il programma.

Questo è il ciclo principale del programma. Mediante un'istruzione di salto condizionato, verifichiamo lo stato del pin RA2 (interruttore SW3). Se l'interruttore è a '1' andiamo alla routine ACCENDERE mediante la quale invieremo un segnale '1' al piedino RB0. Se l'interruttore è a '0' spegneremo il diodo LED mediante la routine SPEGNERE. Questo ciclo è infinito, e viene eseguito in modo continuativo.

```

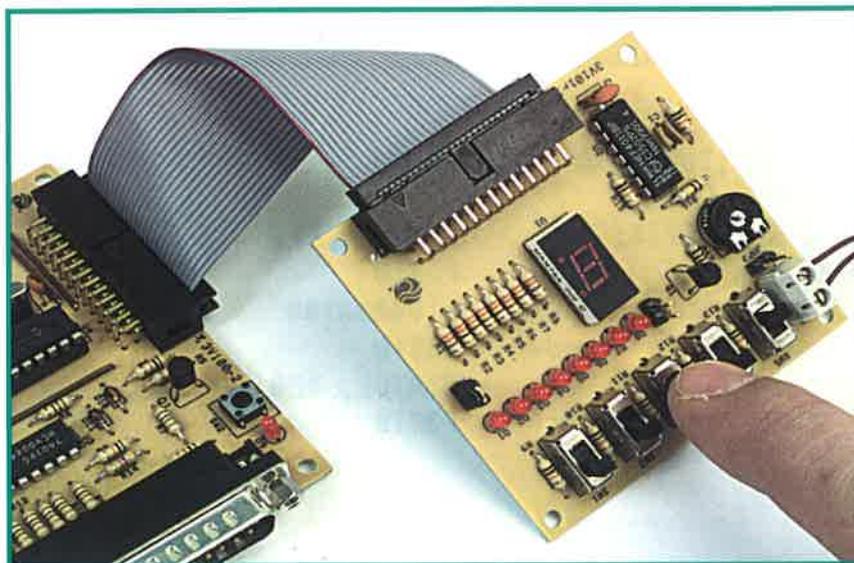
c:\progra~1\mplab\esempio1.asm
32
33 CICLO:      btfss   PORTA, 2
34             goto    SPEGNERE
35             goto    ACCENDERE
36
37 ACCENDERE:  bsf     PORTB, 0
38             goto    CICLO
39
40 SPEGNERE:   bcf     PORTB, 0
41             goto    CICLO
42
43             END     ;Fine del programma
44

```

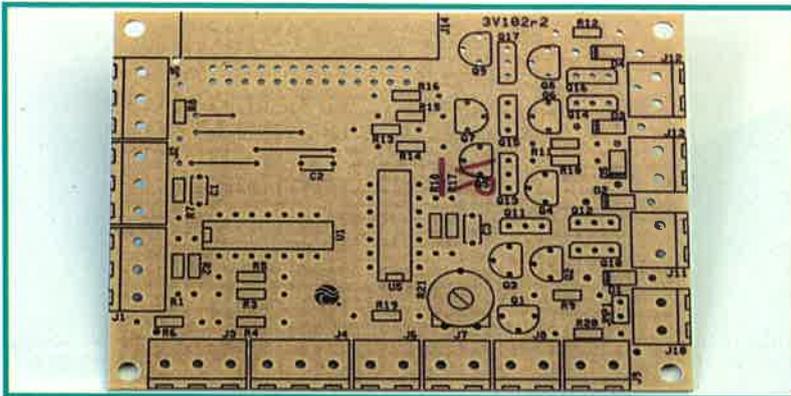


Dopo aver scritto e compilato il programma, procederemo alla memorizzazione del file .hex che è stato generato, nella memoria Flash del microcontroller. Dopo aver scritto e verificato il PIC, procederemo a collegare la scheda di controllo attraverso il connettore PicBus, come ci mostra l'immagine.

Per provare l'esecuzione del programma, dobbiamo togliere tutti i jumper della scheda di ingressi e uscite eccetto JMP1. Questo è quello che abilita i diodi LED, dei quali stiamo utilizzando D1. A seconda di come cambieremo la posizione dell'interruttore SW3 faremo accendere o spegnere il diodo. Sugeriamo ai lettori di ripetere l'esercizio, però provando l'effetto contrario, cioè quando l'interruttore è su ON il LED si dovrà spegnere, e quando è su OFF si dovrà illuminare.

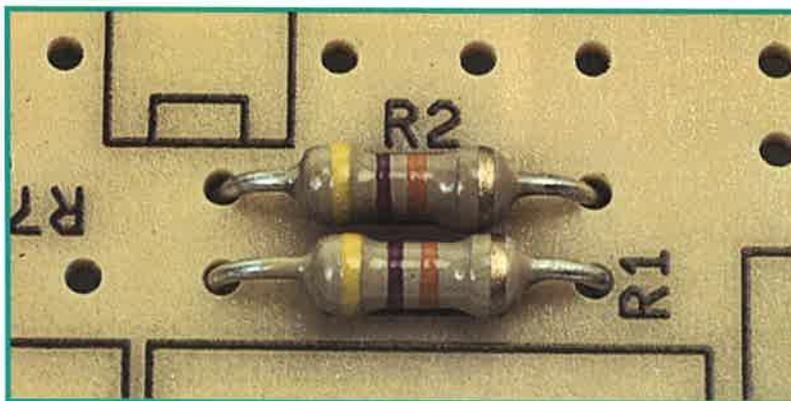
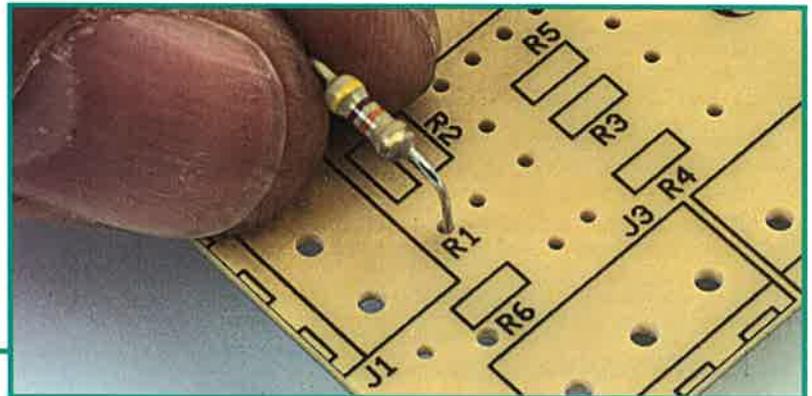


Montaggio passo a passo



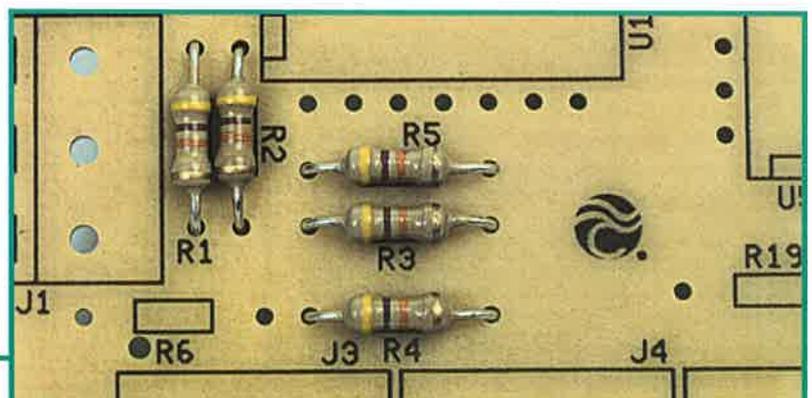
Questa è la scheda di potenza, comunicherà con la scheda di controllo attraverso il PicBus, come quella degli ingressi e uscite; avrà il compito di fornire la potenza necessaria per la gestione dei motori, ed anche di condizionare i segnali di ingresso in arrivo da diversi sensori.

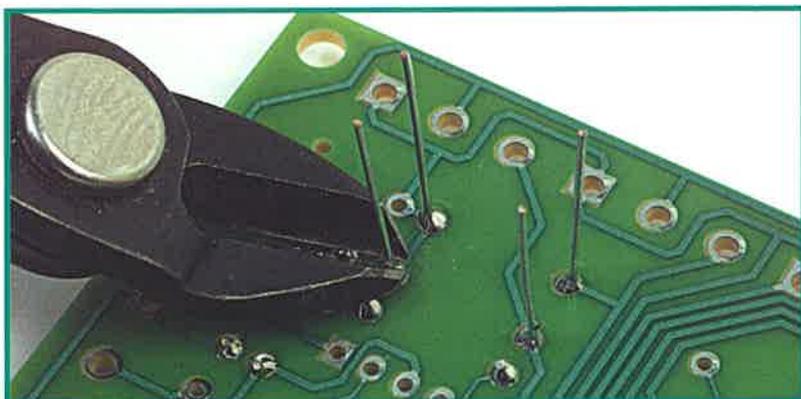
Così come le altre due schede che abbiamo già montato, inizieremo saldando su di essa i diversi componenti, rispettando al massimo le norme che già conosciamo. Inizieremo saldando la resistenza R1 da 47 K (giallo-viola-arancio).



Proseguiremo saldando R2, anch'essa da 47 K. Insistiamo ancora una volta sul fatto che le resistenze dopo che sono state piegate, devono essere inserite a filo della scheda.

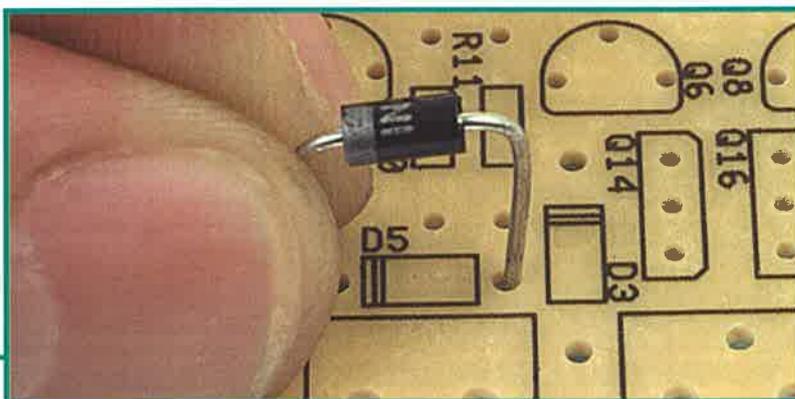
Terminiamo di saldare le resistenze montando R3, R4 e R5 da 47 K ciascuna. Anche se non è obbligatorio, è una buona norma estetica fare in modo che tutte le resistenze abbiano lo stesso orientamento.



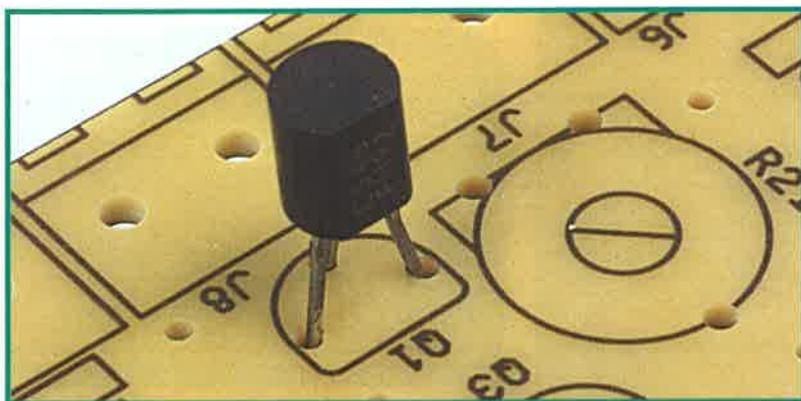


Con l'aiuto di un tronchesino, taglieremo le parti dei reofori che risulteranno in eccedenza. Ricordiamo che il taglio deve realizzarsi a filo del cono di saldatura, appena sopra di esso, ma senza danneggiare la saldatura.

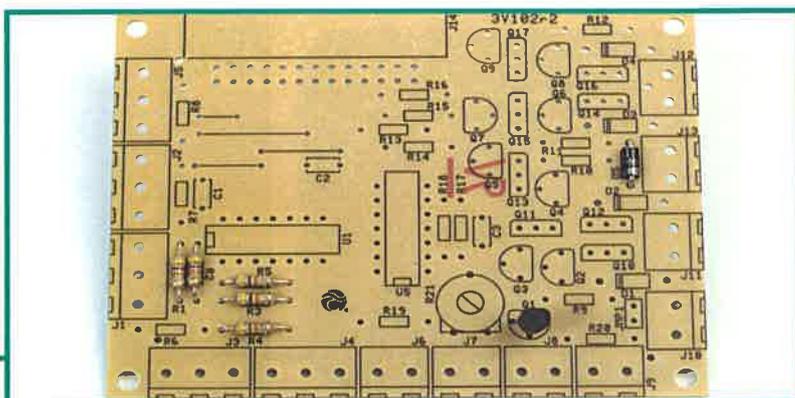
In seguito salderemo il diodo 1N4007, nel posto indicato come D5. Il diodo ha una sua polarità, indicata dalla banda bianca che rappresenta il catodo. Il diodo deve essere orientato facendo coincidere la banda bianca con quella disegnata sulla serigrafia. Si inserisce a filo della scheda, si salda e si raso la rimanenza dei reofori.

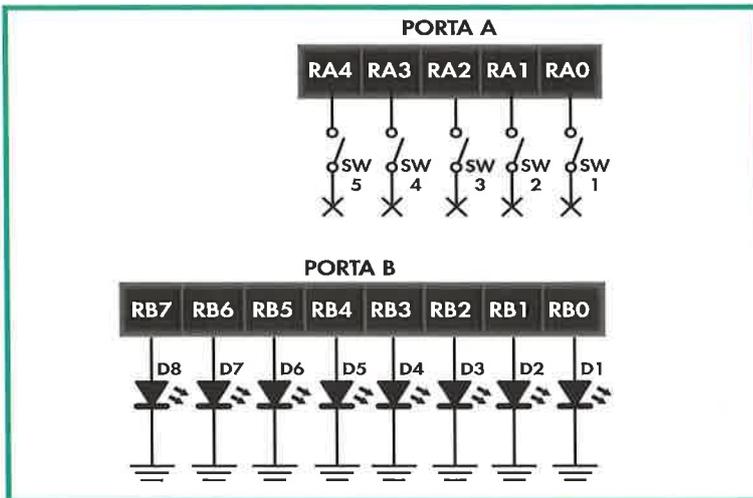


Ora salderemo il transistor Q1, modello BC548. La serigrafia della scheda rappresenta il corretto orientamento; i transistor non si inseriscono a filo della scheda, ma a circa 5 mm di altezza. Si saldano i piedini e poi si taglia la parte in eccesso.



Questo è l'aspetto finale che deve avere la scheda di potenza dopo aver saldato su di essa questi primi componenti. Tutti i componenti devono essere montati al posto che compete loro, e con il corretto orientamento.





Questo esercizio ci permetterà di lavorare con segnali multipli di ingresso e uscita, utilizzando maschere per selezionare insiemi di bit e verificarne il loro valore. Questo programma utilizza i cinque interruttori della scheda di ingressi e uscite, e gli otto diodi LED. All'inizio del programma i diodi rimangono spenti, fino a che viene introdotta la combinazione 1-0-1-0-1 mediante gli interruttori da SW1 a SW5. In quel momento la barra dei LED si illumina. Per spegnerla e tornare ad eseguire il programma, schiacteremo il pulsante di Reset della scheda di controllo.

Nell'intestazione del programma, è buona norma descrivere brevemente, a modo di commento (anteponendo il ;), il compito del medesimo. In seguito selezioneremo il PIC16F84, e creeremo le etichette da quelle dei registri che utilizzeremo nel programma. L'inizio del programma è all'indirizzo 0 e saltiamo il Vector 4 di interrupt.

```

c:\progra~1\mplab\esempio2.asm
1 ;*****
2 ;Programma che attende una combinazione
3 ;degli interruttori di ingresso SW1-SW5
4 ;per accendere tutti i led : D1-D8.
5 ;La combinazione da introdurre è 1-0-1-0-1.
6
7     list p=16F84    ;Tipo di dispositivo
8
9 ;Area delle etichette
10 STATUS EQU    0x03    ;Registro di stato
11 PORTA  EQU    0x05    ;Porta A
12 PORTB  EQU    0x06    ;Porta B
13 RP0    EQU    0x05
14 Z      EQU    0x02
15
16     ORG    0        ;Vector Reset
17     goto  INIZIO
18     ORG    5        ;Saltiamo il Vector di
19                     ;interrupt
20

```

```

c:\progra~1\mplab\esempio2.asm
16     ORG    0        ;Vector Reset
17     goto  INIZIO
18     ORG    5        ;Saltiamo il Vector di
19                     ;interrupt
20
21 ;Inizio del programma
22
23 INIZIO bsf    STATUS, RP0
24       clr   PORTB
25       movlw b'00011111'
26       movwf PORTA
27       bcf   STATUS, RP0
28       clr   PORTB
29

```

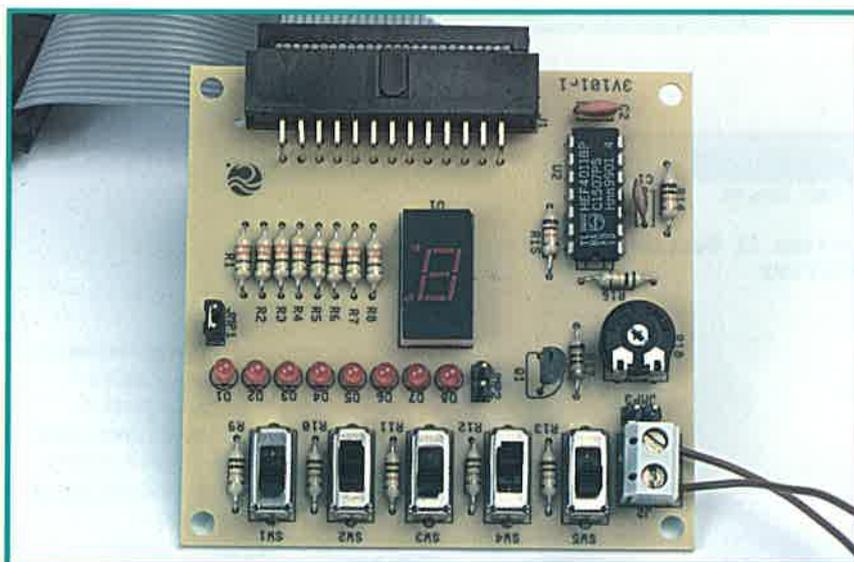
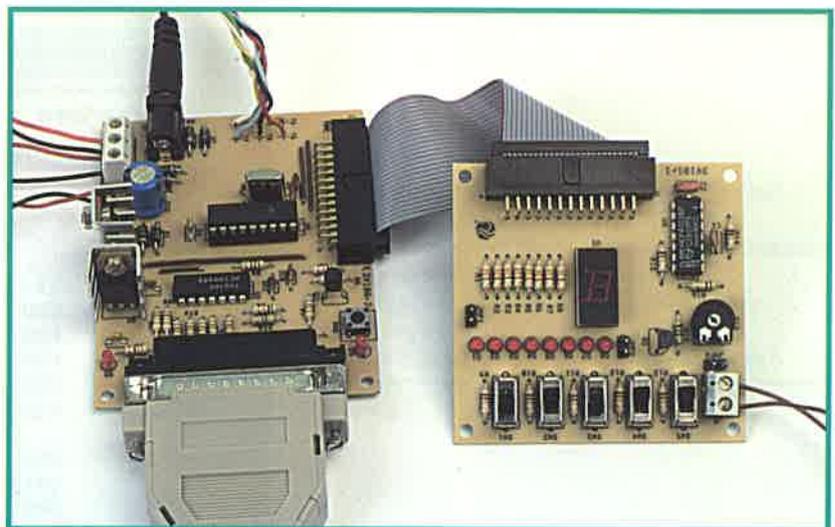
All'inizio del programma configureremo i registri che dovremo utilizzare. Passiamo al banco 1 e in esso configuriamo le porte. La porta A funziona come ingresso, la porta B come uscita. Una volta realizzate le configurazioni torniamo al banco 0, poniamo a zero il valore della porta B, per far sì che all'inizio dell'esecuzione del programma i diodi LED siano spenti.

```

c:\progra~1\mplab\esempio2.asm
31 ;Ciclo che attende la combinazione 1-0-1-0-1.
32
33 CICLO:  movf    PORTA, W
34         andlw  b'00011111'
35         xorlw  b'00010101'
36         btfss STATUS, 2
37         goto  CICLO
38
39         movlw  0xFF
40         movwf  PORTB
41
42 FINE:   goto  FINE
43
44 ;Premere il Reset
45 ;per tornare ad eseguire il programma
46
47         END    ;Fine del programma
48
49 ;*****
    
```

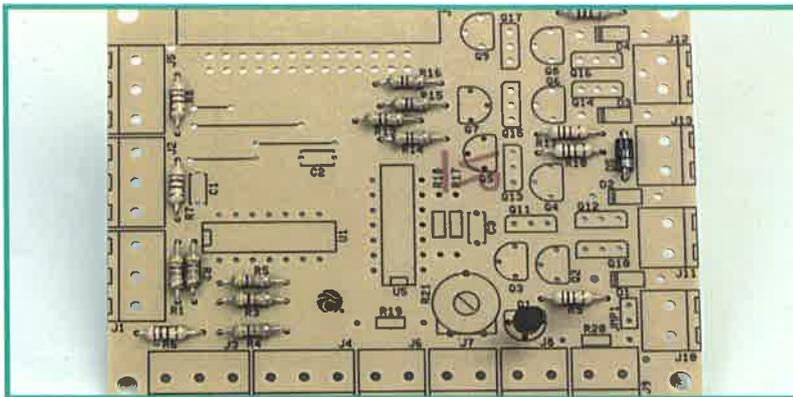
Ciclo principale del programma. In esso leggeremo il valore degli interruttori (Porta A) e realizzeremo una prima maschera AND, per rimanere solo con i cinque meno significativi che vogliamo verificare. La funzione logica XOR verifica che la combinazione introdotta sia valida. Se il risultato delle XOR pone il bit Z del registro di stato a 1 significa che gli interruttori stanno introducendo la combinazione 1-0-1-0-1 e, pertanto, procederemo ad accendere la barra dei LED. In caso contrario, rimarremo all'interno del ciclo principale del programma.

Queste sono le connessioni che dobbiamo realizzare per poter fare questo esercizio. Dobbiamo alimentare la scheda di controllo e collegarla alla porta parallela per la scrittura del file .hex (ottenuto dopo la compilazione) nella memoria Flash del PIC. Ricordiamo che per poter scrivere, il commutatore di scrittura deve essere nella posizione PROG. Infine, dobbiamo collegare le schede di controllo e di ingresso e uscita tramite il connettore PicBus.



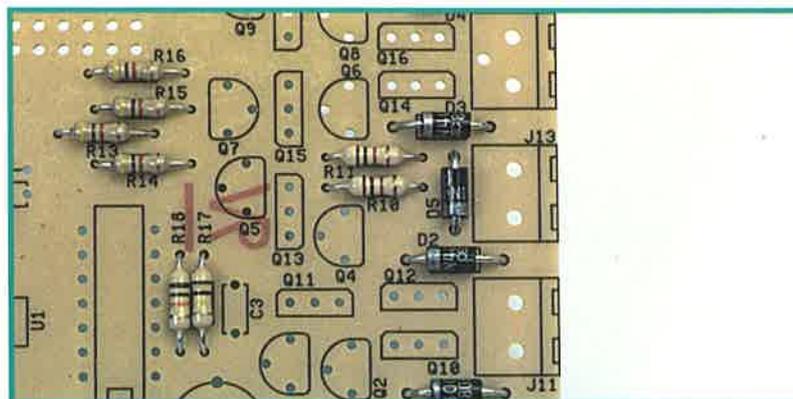
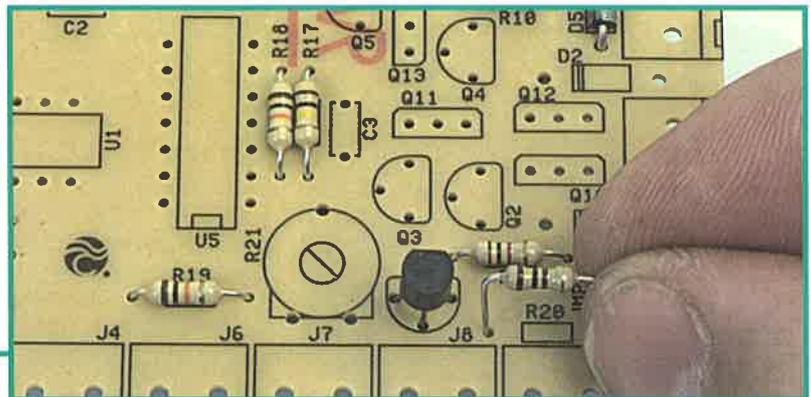
A fianco è mostrato il programma in esecuzione, con la barra dei LED illuminata a conseguenza dell'introduzione della combinazione corretta: 1-0-1-0-1. Per fare in modo che si accendano i LED, il jumper JMP1 deve essere montato, i jumpers JMP2 e JMP3 devono essere estratti e il commutatore di scrittura nella posizione RUN. Questo programma, inoltre, ci servirà per verificare se abbiamo saldato bene tutti i diodi LED della scheda di ingressi e uscite.

Montaggio passo a passo



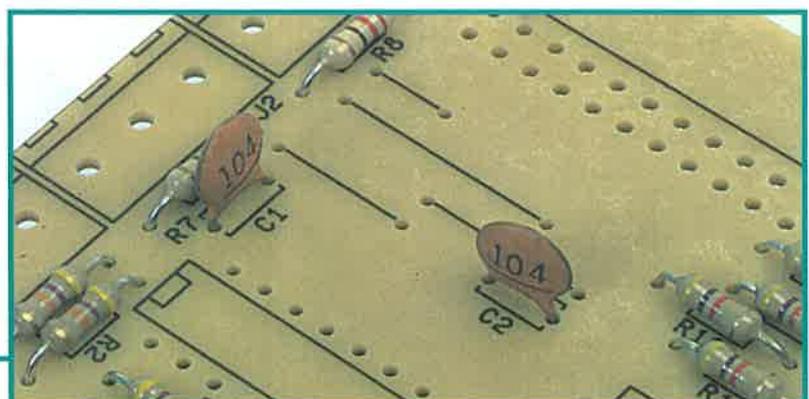
Salderemo tre resistenze da $220\ \Omega$ (rosso-rosso-marrone) denominate R6, R7 e R8, altre quattro resistenze da 1 K (marrone-nero-rosso) che saranno collocate in R9, R10, R11 e R12, ed infine altre quattro da 4K7 (giallo-viola-rosso) ubicate in R13, R14, R15 e R16. Tutte le resistenze devono essere inserite a filo della scheda, e dopo averle saldate, taglieremo le parti dei reofori in eccesso.

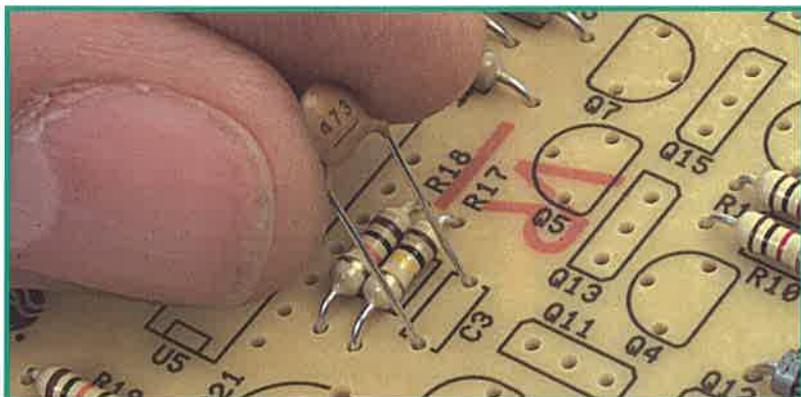
Con la saldatura di altre 4 resistenze si completa la scheda. R17 da 100 K (marrone-nero-giallo), R18 e R19 da 10 K (marrone-nero-arancio), e per ultimo R20 da $100\ \Omega$ (marrone-nero-marrone). Anche in questo caso andranno inserite a filo della scheda ed elimineremo le parti eccedenti dei reofori.



Monteremo i diodi 1N4007 nelle loro sedi, indicate come D1, D2, D3 e D4. Dovremo rispettare l'orientamento degli stessi, facendo coincidere la loro banda bianca con quella serigrafata sulla scheda.

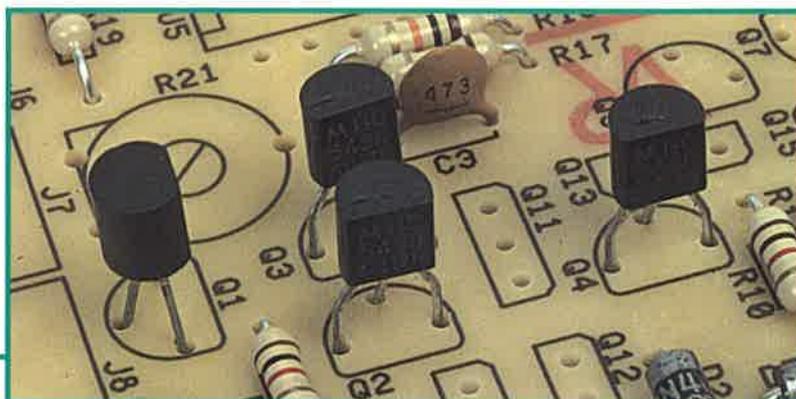
I condensatori C1 e C2 da 100 nF devono essere inseriti a filo della scheda. Non hanno orientamento determinato. Una volta saldati taglieremo i piedini che avanzano.



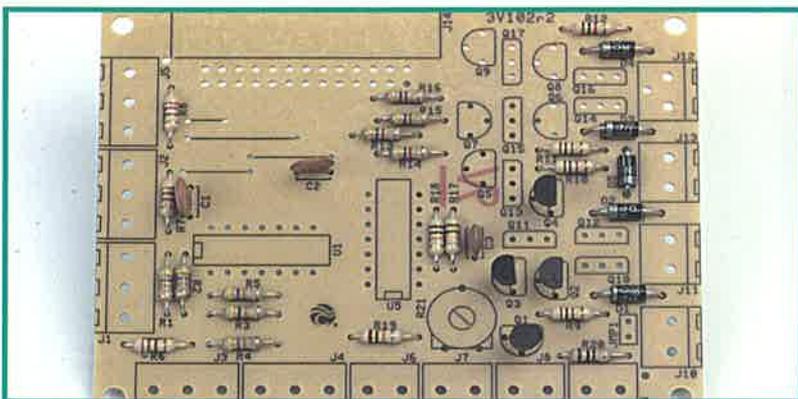


Collochiamo ora il condensatore C3 da 47 nF di capacità. Non ha polarità e pertanto si può inserire in qualunque verso. Si introduce a filo della scheda, si salda e si tagliano i piedini in eccesso.

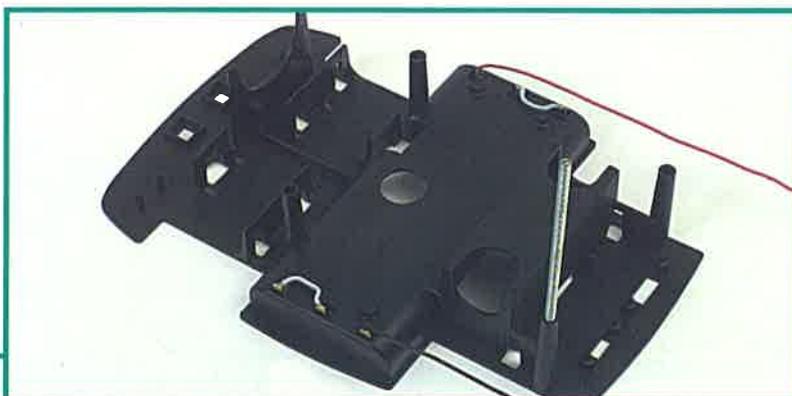
Procederemo saldando i tre transistor modello BC547 che alloggiato nei posti indicati come: Q2, Q3 e Q4. Vengono inseriti a circa 5 millimetri di altezza rispetto al circuito stampato; il disegno della serigrafia ci indica l'orientamento corretto. Per ragioni estetiche dobbiamo far sì che siano tutti allo stesso livello. Una volta saldati, accorceremo la parte dei piedini che fuoriescono dalle saldature.

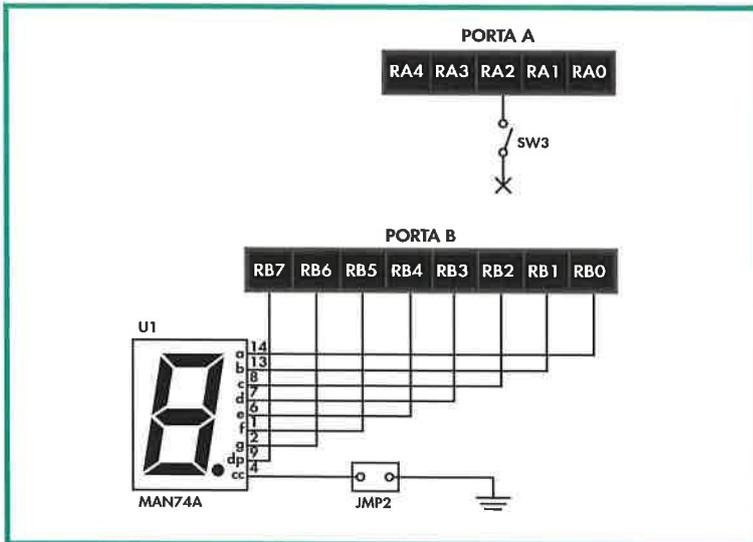


Vista di insieme di come apparirà la scheda di potenza dopo aver saldato tutti i componenti descritti fino ad ora. Faremo una completa revisione per assicurarci che tutto sia corretto.



Nei materiali che vengono forniti, abbiamo una barra filettata che servirà, insieme ad altre tre, per montare le distinte schede sul telaio di Monty. Nell'immagine è mostrata la collocazione della stessa. Va fissata con una goccia di colla.





Con questo esercizio vogliamo introdurre la gestione di un display a sette segmenti. Un display, non è altro che un insieme di diodi LED distribuiti in modo da imitare la forma dei numeri decimali. Rappresentiamo sopra il display della scheda di ingressi e uscite il valore dell'interruttore SW3. Quando questo interruttore è a '0', sul display si vedrà il numero zero, e quando è a '1' verrà mostrato un uno nel display.

Intestazione del programma, dove si realizza una breve descrizione sotto forma di "commento". Dobbiamo scegliere il modello di microcontroller che useremo ed etichettare i registri che saranno impiegati durante lo sviluppo del programma. Infine collocheremo l'inizio del programma nella posizione 0, però salteremo il vector di interrupt 4 prima di introdurre la prima istruzione.

```

c:\progra~1\mplab\esempio3.asm
1 ;*****
2 ;Programma che utilizza il Display a 7 segmenti
3 ;per indicare il valore dell'interruttore SW3.
4
5         list p=16F84      ;Tipo di dispositivo
6
7 ;Area delle etichette
8 STATUS EQU    0x03      ;Registro di stato
9 PORTA  EQU    0x05      ;Porta A
10 PORTB EQU    0x06      ;Porta B
11 RP0   EQU    0x05
12
13         ORG     0        ;Vector Reset
14         goto   INIZIO
15         ORG     5        ;Saltiamo il Vector di
16                               ;interrupt
    
```

```

c:\progra~1\mplab\esempio3.asm
12
13         ORG     0        ;Vector Reset
14         goto   INIZIO
15         ORG     5        ;Saltiamo il Vector di
16                               ;interrupt
17
18 ;Inizio del programma
19
20 INIZIO: bsf     STATUS, RP0
21         cllf   PORTB
22         movlw  b'00000100'
23         movwf  PORTA
24         bcf   STATUS, RP0
25
    
```

Dobbiamo configurare le porte A e B. Per questo puntiamo il banco 1 e poniamo a '1' il bit RP0 del Registro di stato. Una volta in questo banco collocheremo un '1' in RA2 e tutti zero nella porta B. In questo modo la porta B viene configurata come uscita e RA2 funziona come ingresso. Configurate le porte, ritorniamo al banco 0 e poniamo a zero il bit RP0 del Registro di stato.

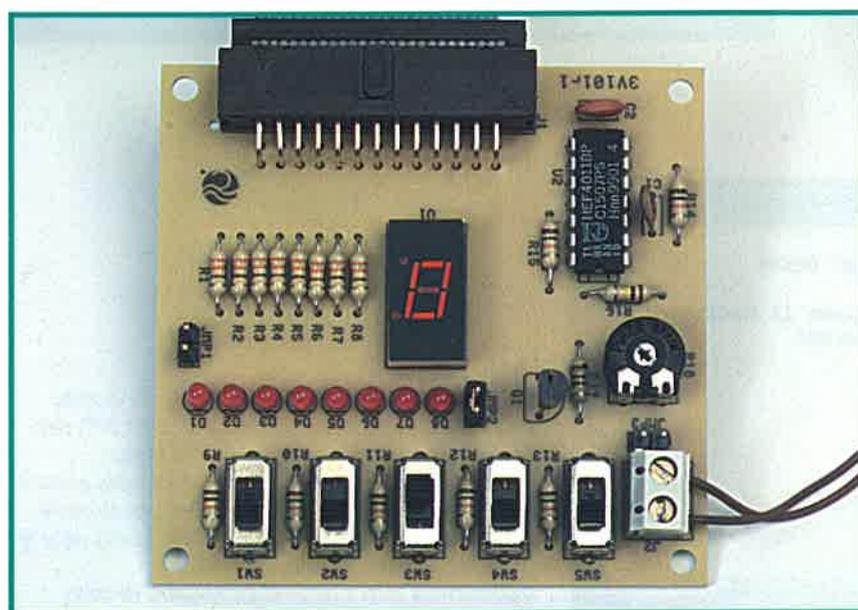
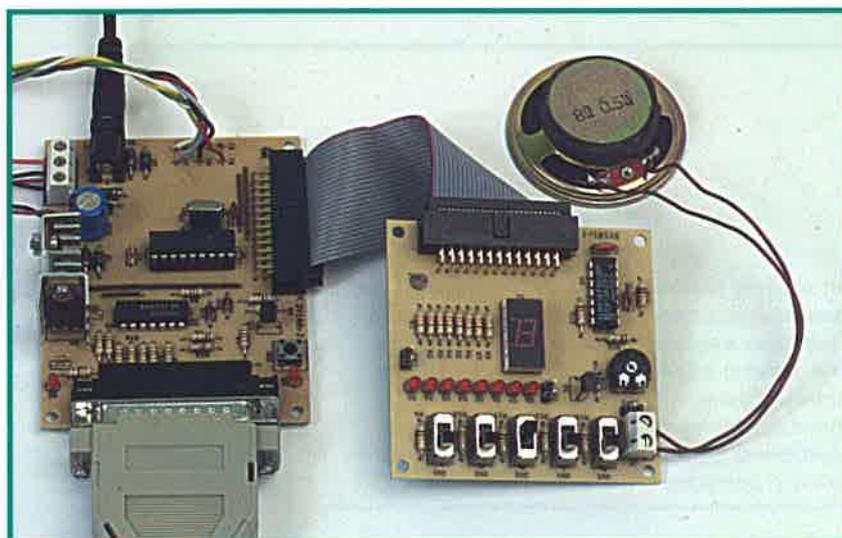
```

c:\progra~1\mplab\esempio3.asm
26
27 CICLO:  btfss   PORTA, 2
28         goto   ZERO
29         goto   UNO
30
31 ZERO:   movlw   b'00111111'   ;Codice 0
32         movwf  PORTB
33         goto   CICLO
34
35 UNO:    movlw   b'00000110'   ;Codice 1
36         movwf  PORTB
37         goto   CICLO
38
39         END       ;Fine del programma
40 ;*****

```

Questo è il ciclo principale del programma, il quale viene eseguito in maniera continuativa. In esso testiamo il valore dell'interruttore SW3, mediante la lettura del pin RA2. Se è a zero, andiamo alla routine con la quale mandiamo attraverso la porta B il codice a sette segmenti corrispondente allo zero, in modo che sia mostrato nel display. Se l'interruttore è a '1', saltiamo alla routine UNO con la quale mandiamo al display il codice corrispondente al numero '1'.

Una volta scritto e compilato il programma, procederemo a memorizzarlo nel microcontroller. Per fare questo collegheremo la scheda di controllo al PC mediante la porta LPT1 e ci assicuriamo che l'interruttore di memorizzazione sia in PROG. Dobbiamo collegare la scheda di controllo e quella degli ingressi e uscite mediante il PicBus, per verificare l'esecuzione del programma.

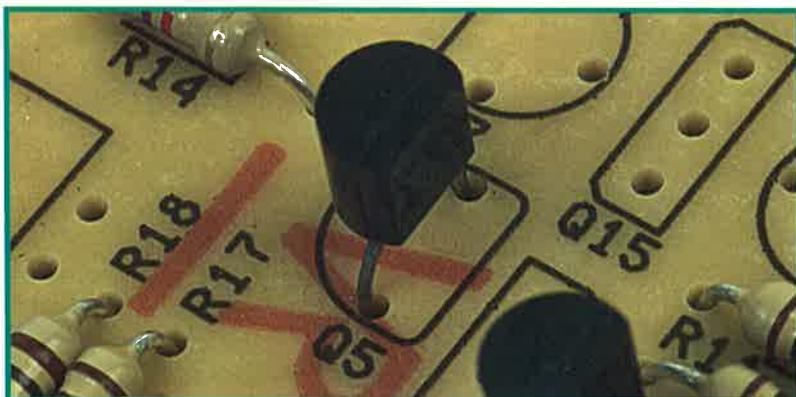


Qui si vede il display che sta visualizzando uno '0', il cui valore corrisponde a quello presente sull'interruttore SW3. Per gestire il display è necessario chiudere il jumper JMP2 e aprire JMP1 e JMP3, come si vede nell'immagine. L'interruttore della scheda di controllo, ora deve essere collocato in RUN invece che in PROG.

MODULO DI POTENZA

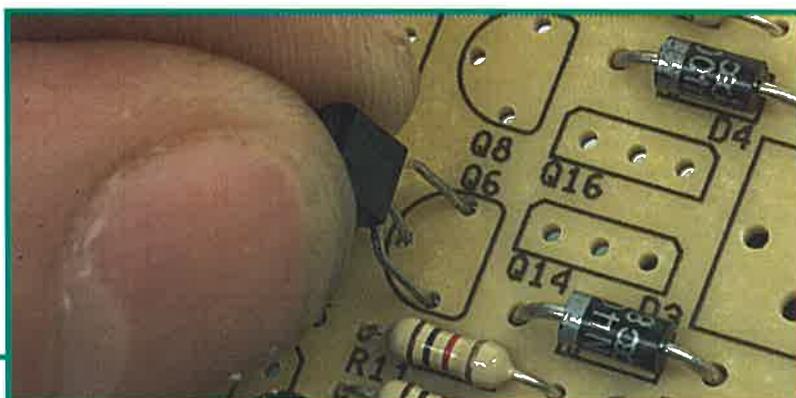
PL 46

Montaggio passo a passo

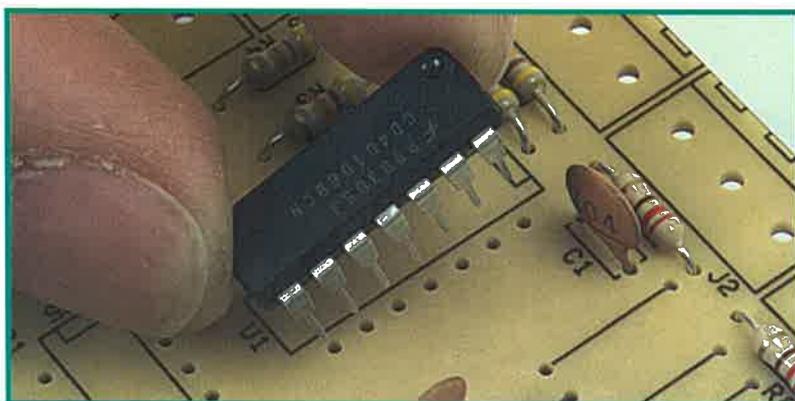


Il transistor Q5, modello BC547, si inserisce a circa 5 mm di altezza rispetto alla scheda. Il disegno della serigrafia ci fornisce informazioni precise su come deve essere orientato. Una volta saldato taglieremo la parte dei piedini in eccesso.

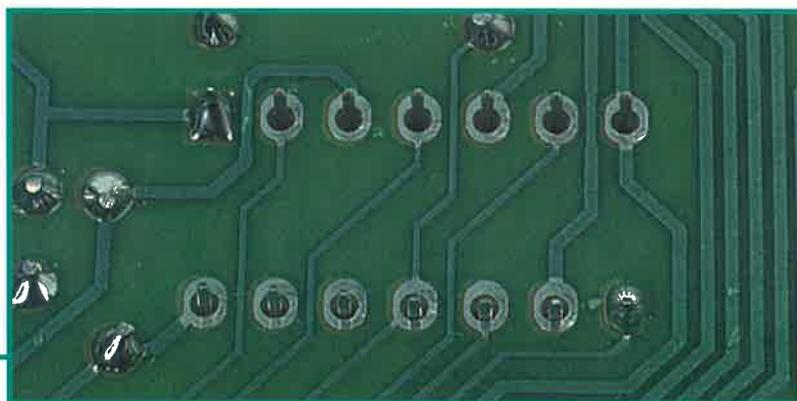
Ora monteremo il transistor Q6, modello BC547, che come il precedente, deve essere inserito a 5 mm di altezza rispetto alla scheda. L'orientamento è indicato nel disegno della serigrafia. Inoltre dobbiamo eliminare la parte dei terminali che risulterà in eccesso una volta saldati.

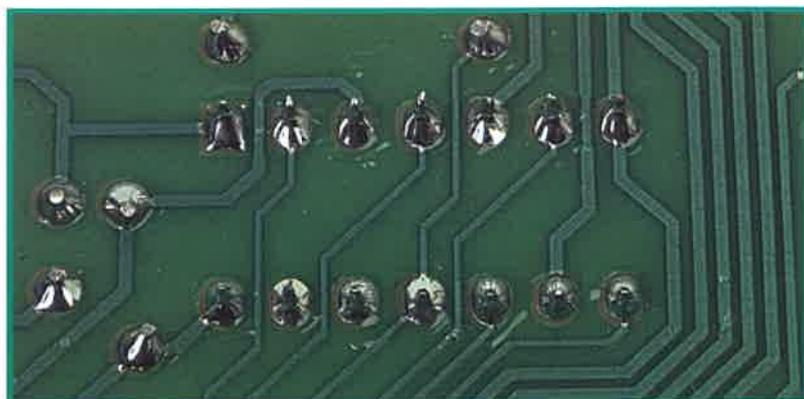


Collocheremo il circuito integrato modello 40106 nel posto indicato come U1. Si tratta di un circuito di condizionamento del segnale logico. Dobbiamo fare molta attenzione all'orientamento del medesimo: la sua tacca di riferimento deve coincidere con quella indicata nella serigrafia. Inserendolo, dobbiamo assicurarci di non piegare nessun piedino.



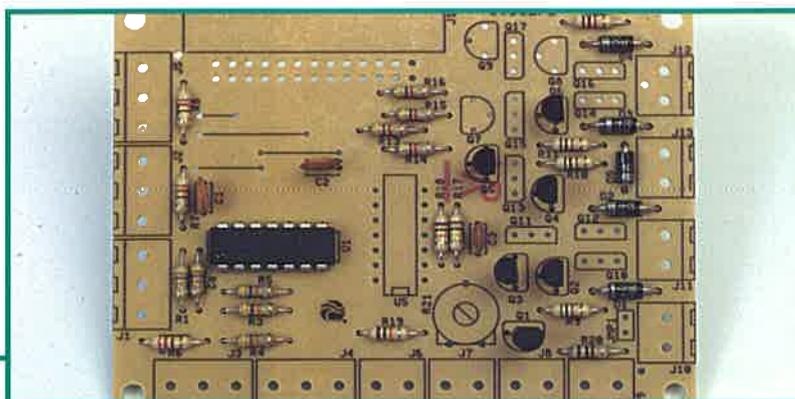
Per prima cosa, così come è mostrato nell'immagine, si saldano solo i due piedini agli estremi opposti dell'integrato. In questo modo resta collegato alla scheda.





Quando saremo sicuri che il circuito integrato sia stato inserito con l'orientamento corretto e che non sia stato piegato nessun piedino, procederemo con la saldatura di tutti gli altri piedini. Fra la saldatura di un piedino e un'altra, conviene lasciar passare un po' di tempo in modo che non si accumulino un eccesso di calore che potrebbe danneggiare il chip.

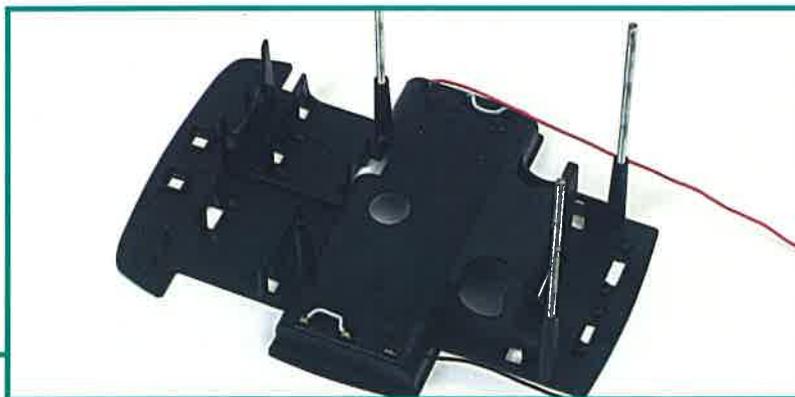
L'immagine ci mostra l'aspetto della scheda di potenza con tutti i componenti saldati fino a questo momento. Controllare che tutti siano correttamente collegati.

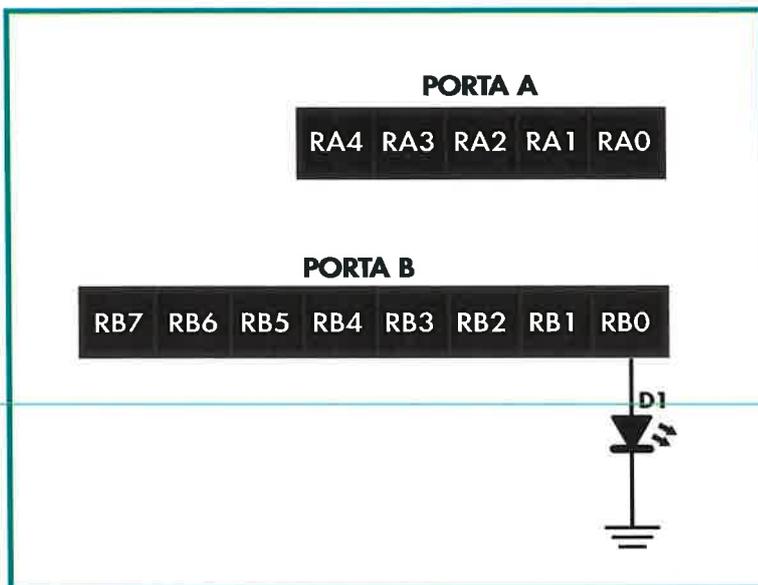


Con questo fascicolo vengono fornite anche due barre filettate come quelle dell'immagine. Una barra è lunga 94 mm e l'altra 107 mm.



Le barre filettate si inseriscono nel telaio di Monty. In seguito serviranno per sorreggere e dividere le diverse schede. Nell'immagine si vede come restano una volta montate. La barra corta è situata nella parte anteriore del telaio e quella più lunga nella parte posteriore; vengono fissate con una goccia di adesivo.





In questo esercizio realizzeremo una temporizzazione. Il programma che implementeremo e proveremo sulla scheda di ingressi e uscite, farà lampeggiare il diodo LED D1 accendendolo e spegnendolo ogni secondo. Nell'immagine è indicato lo schema dei collegamenti necessari per poter provare questo programma.

Come già sappiamo all'inizio del programma dobbiamo definire il tipo di microcontroller che vogliamo utilizzare. Inoltre possiamo dare un nome ai registri del PIC che utilizzeremo nel corso del programma. Oltre ai registri interni del PIC, in questo programma utilizzeremo anche un registro di uso generale, che posizioneremo al primo indirizzo libero, che corrisponde allo 0x0C.

```

c:\progra~1\mplab\esempio4.asm
1 ;*****
2 ;Programma che accende e spegne il Diodo D1
3 ;ogni secondo.
4
5     list    p-16F04 ;Tipo di dispositivo
6
7 ;Area delle etichette
8 STATUS EQU    0x03 ;Registro di stato
9 PORTA  EQU    0x05 ;Porta A
10 PORTB EQU    0x06 ;Porta B
11 RP0   EQU    0x05
12 OPTION_REG EQU 0x01 ;Timer0 e Option
13 INTCON EQU    0x0b
14 TEMPO1 EQU    0x0c ;Registro di utilizzo
15                               ;generale
16
17     ORG     0 ;Vector Reset
18
19     goto   INIZIO
20     ORG     5 ;Saltiamo il vector di
21                               ;interrupt.
    
```

```

c:\progra~1\mplab\esempio4.asm
16
17     ORG     0 ;Vector Reset
18
19     goto   INIZIO
20     ORG     5 ;Saltiamo il vector di
21                               ;interrupt.
22
23 ;Inizio del programma
24
25 INIZIO: bsf    STATUS, RP0
26         clrf  PORTB
27         movlw b'00000111'
28         movwf OPTION_REG
29         bcf  STATUS, RP0
30         clrf  PORTB
    
```

All'inizio del programma passeremo al banco 1 per realizzare le configurazioni. Imposteremo la Porta B come uscita, e assegneremo il prescaler al Timer0, con valore 256. Questo farà sì che il Timer0 si incrementi ogni 256 cicli macchina, e ci faciliterà i calcoli, dato che dobbiamo realizzare una temporizzazione così lunga, qual è quella di 1 secondo (bisogna ricordare che il PIC impiega 1 microsecondo ad eseguire un'istruzione). Dopo aver configurato questi registri, torneremo al banco 0.

```

c:\progra~1\mplab\esempio4.asm
31
32
33 ;Ciclo principale. Si accende e si spegne
34 ;il led temporizzando un secondo fra
35 ;entrambe le azioni
36
37 CICLO:  bsf      PORTB, 0
38         call    DELAY1S
39         bcf      PORTB, 0
40         call    DELAY1S
41         goto    CICLO
42
43

```

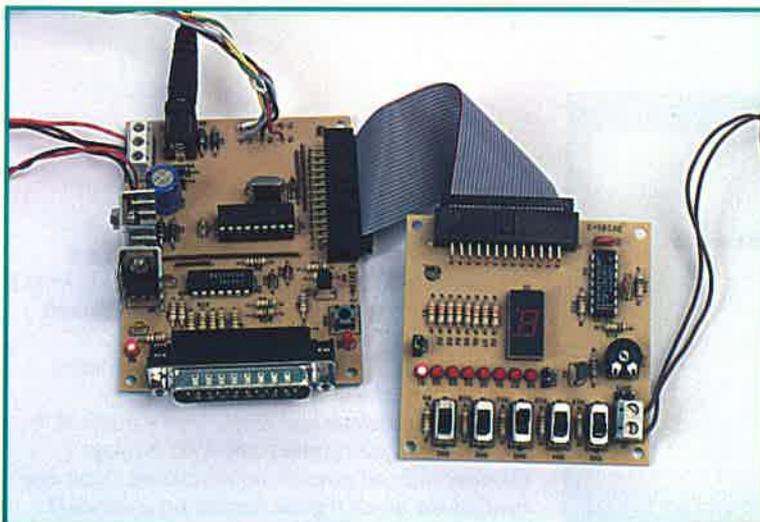
Questo è il ciclo principale del programma, dal quale eseguiamo la chiamata alla routine di configurazione. Il procedimento è il seguente: si accende il diodo, si temporizza un secondo, si spegne il diodo, si temporizza un secondo e il ciclo ricomincia. Questo programma, quindi, rimane in esecuzione in modo permanente.

Questa è la routine di temporizzazione di 1 secondo. Prima si carica il registro TEMPO1 con il valore decimale 100, dopo si chiama un'altra routine che esegue una temporizzazione di 10 ms, un numero di volte pari al valore della variabile TEMPO1. In questo modo otteniamo una temporizzazione di 1 secondo. La routine DEL10 esegue con il Timer0 la seguente temporizzazione:
 $T = 4 * 0.25\mu s * 39 * 256 = 0.009984 \text{ secondi} = 10 \text{ ms (circa)}$

```

c:\progra~1\mplab\esempio4.asm
44 ;Routine di ritardo di 1 secondo
45
46 DELAY1S bcf      STATUS, RP0
47         movlw   0x64
48         movwf  TEMPO1
49         call   DEL10
50         return
51
52 DEL10  bcf      INTCON, 2
53         movlw   0xD8
54         movwf  OPTION_REG
55
56 DEL10_1 btfss   INTCON, 2
57         goto    DEL10_1
58         decfsz TEMPO1, 1
59         goto    DEL10
60         return
61
62         END    ;Fine del programma

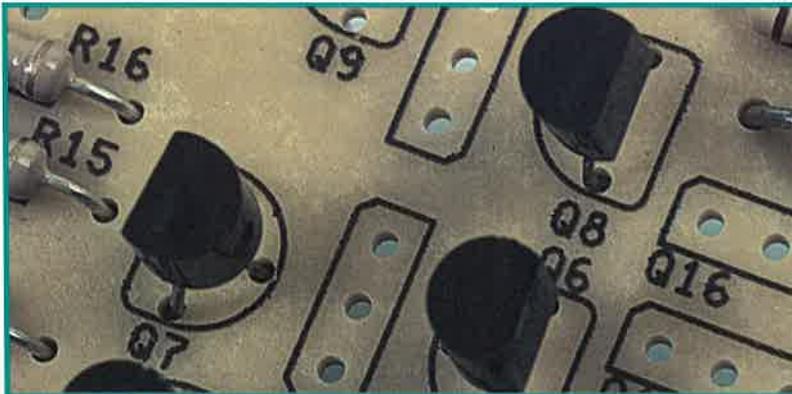
```



Programma in esecuzione sulla scheda di ingressi e uscite, che è collegata alla scheda di controllo mediante il PicBus.

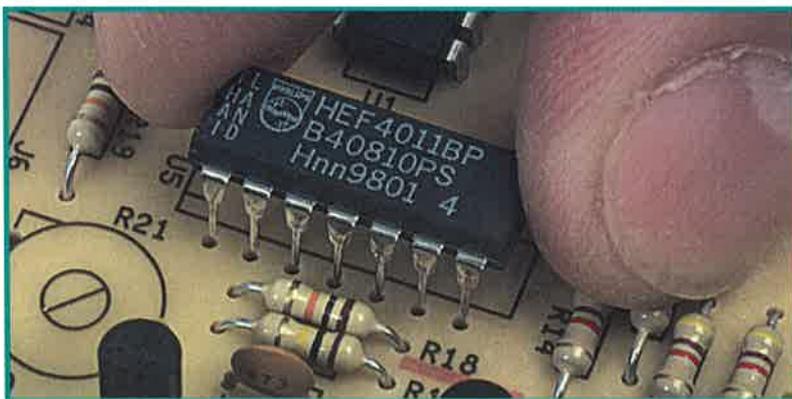
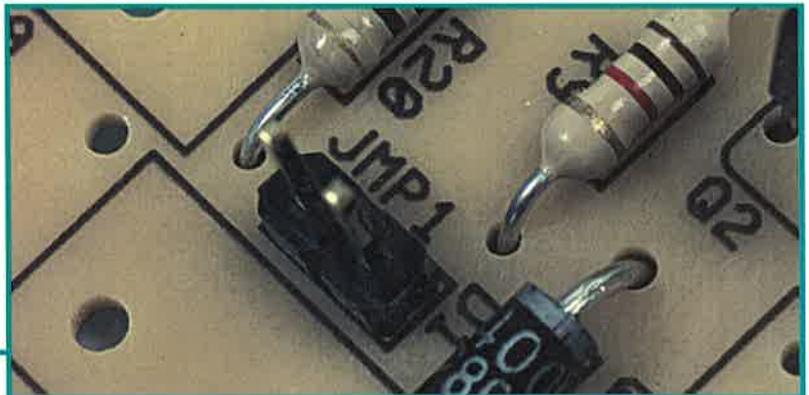
Il jumper JMP1 deve essere chiuso, mentre il resto dei jumpers deve essere aperto. Proponiamo al lettore di cercare di ottenere la stessa temporizzazione di 1 secondo, ma con un prescaler di 128 assegnato al Timer0.

Montaggio passo a passo



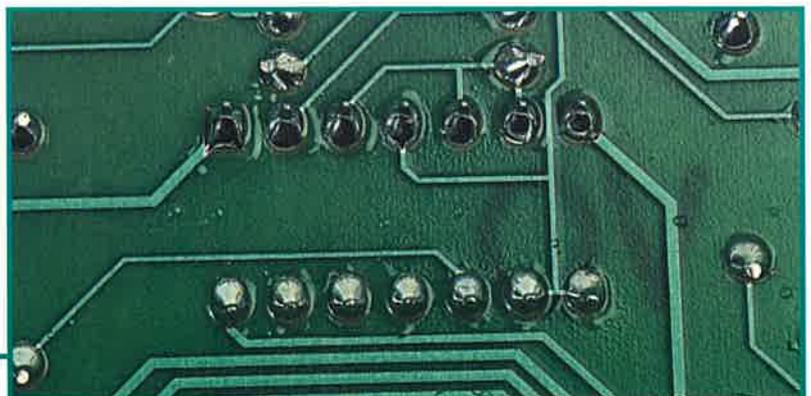
Nelle posizioni destinate a Q7 e Q8, monteremo i due transistor modello BC547. Devono essere inseriti a 5 mm di altezza rispetto alla scheda. Dopo aver saldato i piedini, taglieremo la parte di reoforo che fuoriesce dalla saldatura.

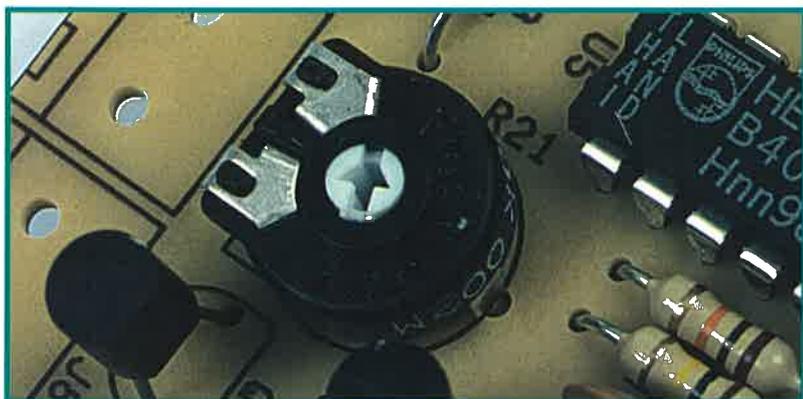
Come possiamo vedere dall'immagine, continueremo il montaggio con il connettore dritto a due posizioni, destinato al jumper JMP1. Il lato corto dei piedini va inserito nella scheda per essere saldato, mentre il lato lungo rimane nella parte superiore.



Monteremo il circuito integrato modello 4011 nel posto riservato per U5. Questo circuito contiene porte logiche di tipo NAND. La tacca di riferimento del medesimo deve essere orientata secondo il disegno della serigrafia. Durante l'inserzione dobbiamo fare attenzione che non venga piegato nessun piedino del chip.

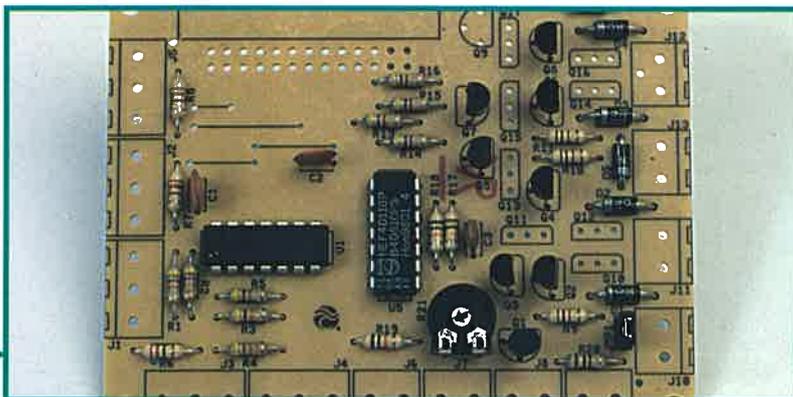
Come abbiamo già visto in precedenza, è consigliabile iniziare la saldatura dai piedini di due estremi opposti; dopo ci assicureremo che il componente sia orientato in modo corretto e che non abbia nessun pin piegato, prima di procedere alla saldatura degli altri piedini, così come mostrato dall'immagine.





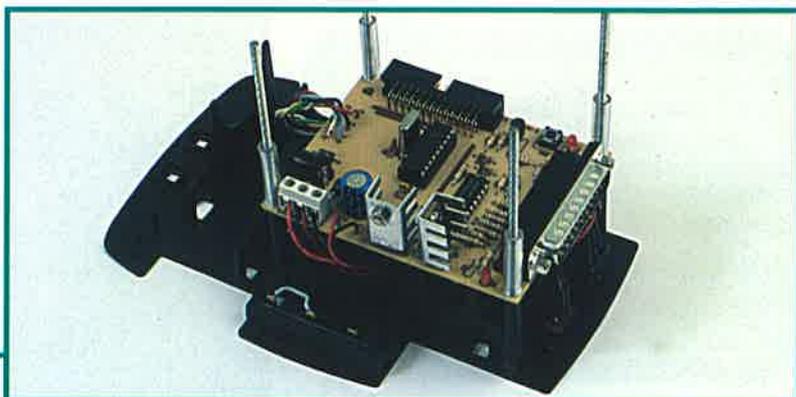
La resistenza R21 è in realtà un potenziometro regolabile, con valore 10 K. Il suo orientamento non deve presentare nessun problema, dato che può essere montato solo in una posizione. Deve essere inserito a filo della scheda.

L'immagine mostra l'aspetto del circuito stampato dopo aver montato tutti i componenti forniti sino a questo momento. Notate che sul connettore JMP1 abbiamo già montato il corrispondente ponticello.



Con questo fascicolo è stata fornita l'ultima barra filettata da 94 mm, insieme ad un tubetto di PVC da 3x5 mm lungo 250 mm. Così come illustrato nell'immagine, dobbiamo tagliarne 4 pezzi di questo tubo lunghi circa 25 mm cadauno.

L'immagine mostra il telaio di Monty con le quattro barre filettate installate per ancorare le diverse schede. Possiamo notare inoltre che i quattro pezzi di tubo di PVC tagliati in precedenza, sono stati inseriti nelle barre per fare da separatori e permettere la sovrapposizione delle distinte schede. Raccomandiamo, comunque, di fare questa operazione più avanti.



```

c:\progra~1\mplab\esempio5.asm
1 ;*****
2 ;Programma contatore
3
4     list    p=16F84 ;Tipo di dispositivo
5
6 ;Area delle etichette
7 STATUS EQU    0x03 ;Registro di stato
8 PORTA EQU    0x05 ;Porta A
9 PORTB EQU    0x06 ;Porta B
10 RP0 equ     0x05
11 OPTION_REG EQU 0x01 ;Timer0 e Option
12 INTCON EQU   0x0B
13 ;Registro di utilizzo generale
14 CONTATORE EQU 0x0D
15
16     ORG     0 ;Vector Reset
17
18     goto   INIZIO
19     ORG     5 ;Saltiamo il vector di
20           ;interrupt.
    
```

Nell'esempio seguente realizzeremo un contatore. Utilizzeremo come ingresso l'interruttore SW1 della scheda di ingressi e uscite, che corrisponde al piedino RA0 del PIC. Il valore contato verrà riportato sui diodi LED da D1 a D8 in forma di numero binario. Verranno contate le transizioni 0-1-0 che saranno inviate all'ingresso tramite il commutatore. La prima cosa che faremo sarà nominare i registri che utilizzeremo nel corso del programma, come abbiamo già visto anche altre volte.

La configurazione del microcontroller richiede di configurare il piedino RA0 come ingresso, per poter leggere il segnale dell'interruttore. La porta B sarà l'uscita per visualizzare il dato del contatore. Dobbiamo configurare un prescaler di 256 per il Timer0, dato che sarà utilizzato per evitare i rimbalzi.

```

c:\progra~1\mplab\esempio5.asm
21
22 ;Inizio del programma
23
24 INIZIO: bsf     STATUS, RP0
25         movlw  0x01
26         movwf  PORTA
27         clrfs  PORTB
28         movlw  b'00000111'
29         movwf  OPTION_REG
30         bcf     STATUS, RP0
31         clrfs  PORTB
32         clrfs  CONTATORE
33
    
```

```

c:\progra~1\mplab\esempio5.asm
33
34 ;Ciclo principale. Si esegue la somma in funzione
35 ;degli impulsi introdotti dall'interruttore SW1
36
37 CICLO:  btfss   PORTA, 0
38         goto   CICLO
39         call   ANTIRIMBALZO
40
41 CICLO2: btfsc   PORTA, 0
42         goto   CICLO2
43         call   ANTIRIMBALZO
44
45         incf   CONTATORE, F
46         movf  CONTATORE, W
47         movwf PORTB
48
49         goto  CICLO
50
    
```

Questo è il ciclo principale del programma, il quale attende che dall'interruttore arrivi una transizione da 0 a 1 e che torni in stato di riposo '0'. In questo istante si realizza la somma, incrementando la variabile contatore. Il risultato della somma viene visualizzato tramite la porta B.

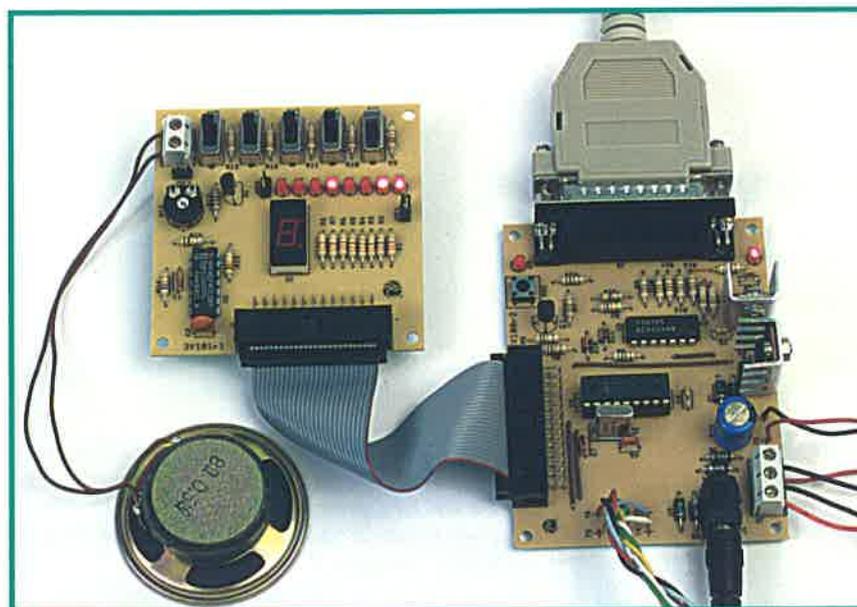
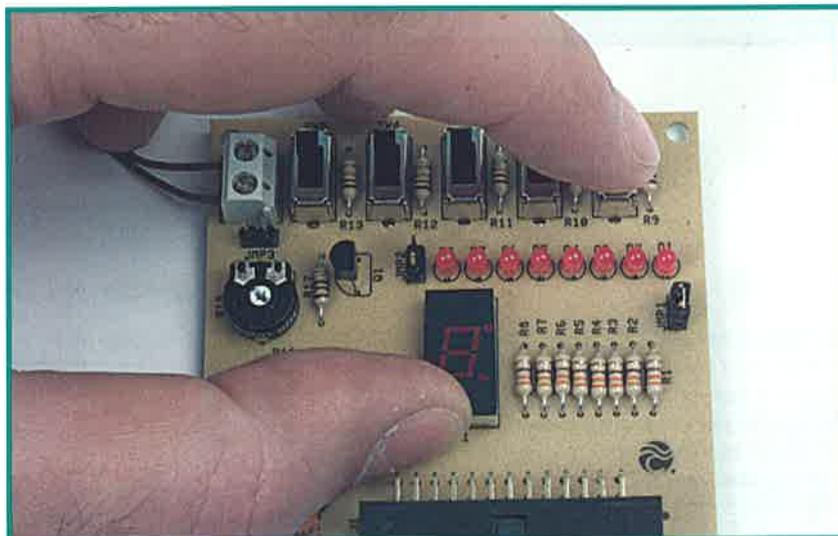
```

c:\progra~1\mplab\esempio5.asm
50
51
52 ;Routine per risolvere via software il problema
53 ;dei rimbalzi degli interruttori.
54
55 ANTIRIMBALZO:
56     bcf     INTCON, 2
57     movlw  0x80
58     movwf  OPTION_REG
59
60 DEL:   btfss  INTCON, 2
61         goto  DEL
62         return
63
64     END    ;Fine del programma
65 ;*****|

```

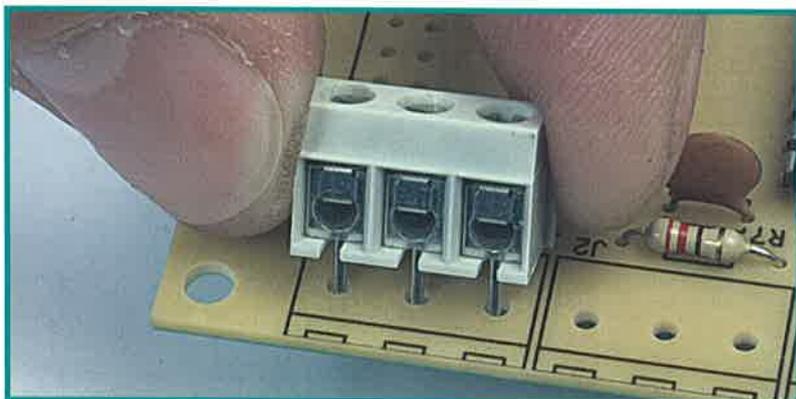
Dato che questo programma utilizza periferiche reali, dobbiamo anche considerare gli effetti intrinseci di questi componenti, in questo caso dobbiamo evitare i rimbalzi del commutatore. L'effetto rimbalzo è uno stato transitorio, che genera una sequenza casuale di zeri e uno che si produce quando un sensore di tipo meccanico cambia di stato. Per evitare i rimbalzi realizzeremo una temporizzazione ogni volta che rileviamo un cambio di stato nell'interruttore.

Nella foto si mostra il funzionamento del programma sulla scheda di ingressi e uscite. Introduciamo gli impulsi attraverso il commutatore SW1, e osserveremo il dato di risposta sui diodi LED. A tal fine dobbiamo tenere il jumper JMP1 chiuso e il resto dei jumpers aperti.



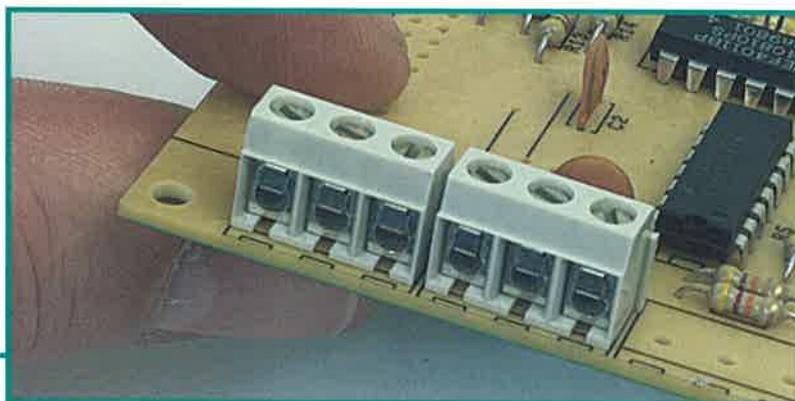
In questa immagine possiamo osservare i collegamenti fra la scheda di controllo e quella di ingressi e uscite. Dopo aver compilato il programma procederemo alla sua scrittura sul microcontroller, tramite la scheda di controllo, nel modo che già conosciamo. Vi consigliamo di ripetere l'esercizio, però annullando la chiamata alla routine ANTIRIMBALZO, e osservare sulla scheda di ingressi e uscite il funzionamento del programma appena modificato.

Montaggio passo a passo

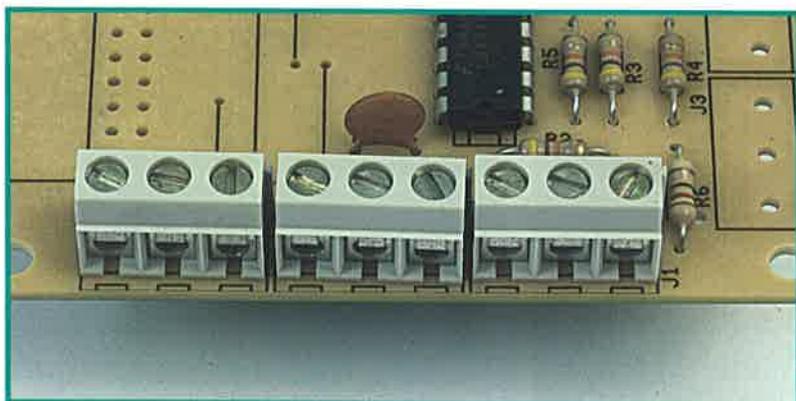


Come prima cosa montiamo una delle morsettiere a tre contatti nella posizione J5. Gli ingressi della morsettiere devono essere orientati verso il lato esterno della scheda di potenza.

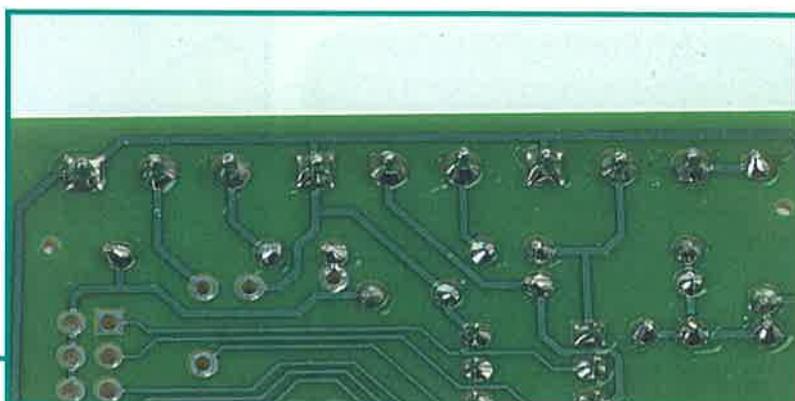
Inseriremo un'altra morsettiere a tre contatti in J2, a lato di quella che abbiamo appena montato. Le morsettiere devono essere inserite completamente, sino ad appoggiare sulla scheda. Il verso di inserzione per le morsettiere sarà sempre con gli ingressi verso l'esterno della scheda.

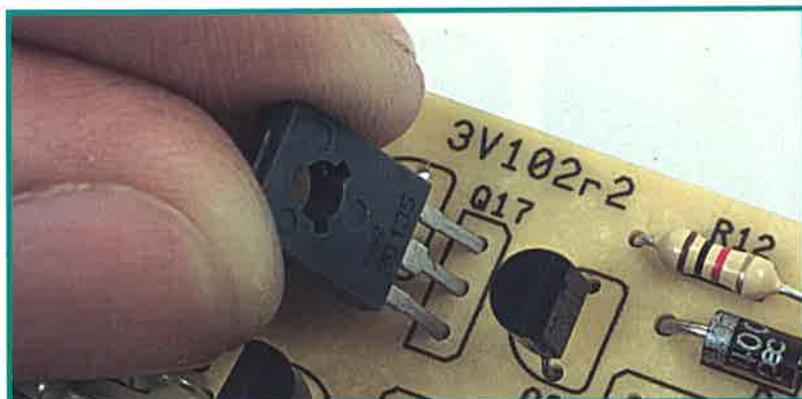


Infine collocheremo l'altra morsettiere a tre contatti in J1. Ora procederemo alla saldatura delle tre morsettiere. Prima di iniziare le saldature dobbiamo verificare che le morsettiere non siano piegate e che appoggino bene sulla superficie della scheda.



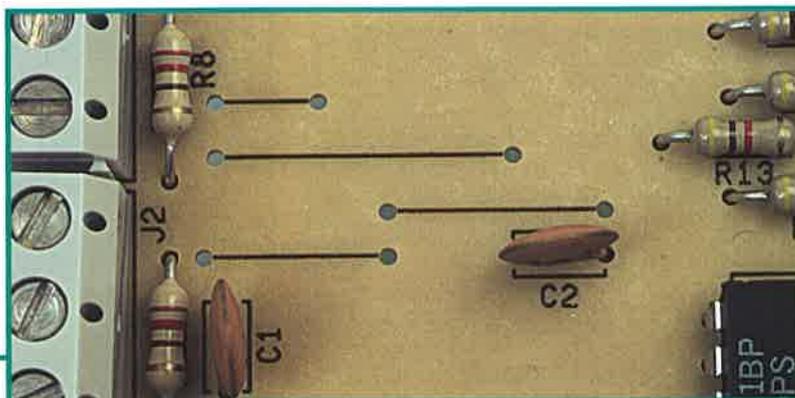
Come si osserva nell'immagine, dobbiamo saldare sino a che lo stagno non abbia riempito tutta la piazzola della morsettiere. La quantità di stagno utilizzata per queste saldature è maggiore di quella che abitualmente serve per il resto dei componenti della scheda.



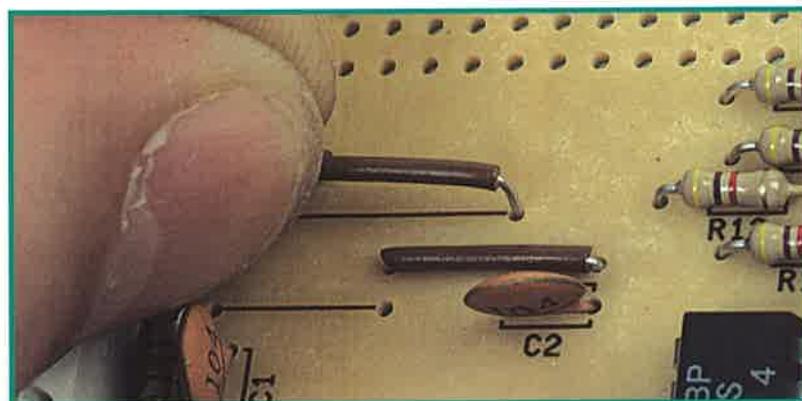


Il prossimo componente da montare è un transistor modello BD135, lo inseriremo in Q17. Dobbiamo prestare particolare attenzione al verso d'inserzione del transistor, il quale deve essere come quello mostrato nell'immagine. Non dobbiamo introdurre i piedini del transistor sino in fondo, ma deve rimanere sollevato dalla scheda alcuni millimetri.

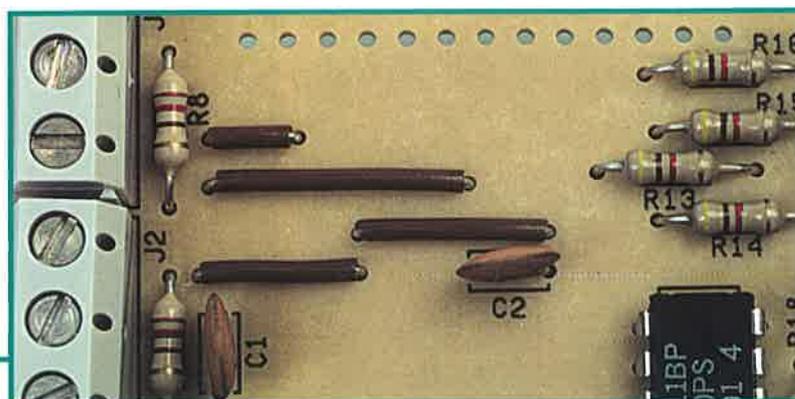
Ora dobbiamo realizzare una serie di ponticelli, utilizzando il cavo da 0,5 mm. I ponticelli si trovano situati nella parte della scheda che si vede nell'immagine, sono indicati da una linea nera che unisce due fori.

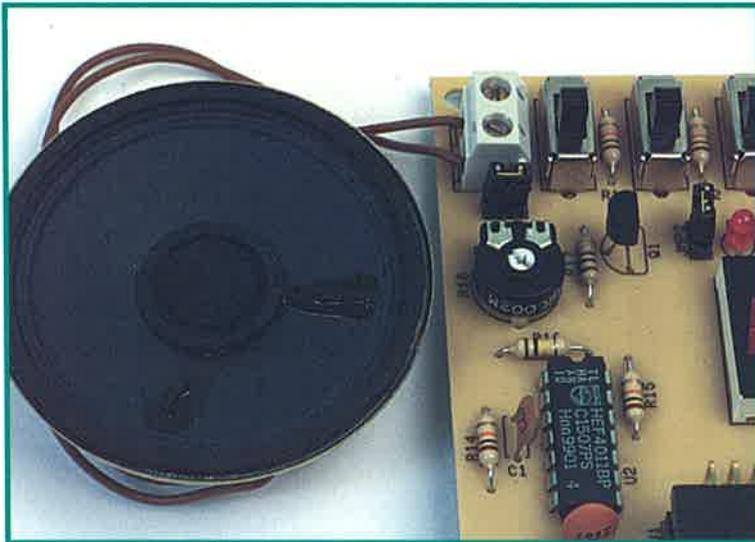


Taglieremo i pezzi di filo della dimensione adatta ad ogni ponticello, speleremo ogni pezzo da entrambe le estremità, e lo inseriremo come mostrato nell'immagine. In seguito salderemo i pezzi di cavo alla scheda.



Ecco come rimangono i quattro ponticelli della scheda di potenza, i cavetti devono essere saldati in modo corretto, grazie a loro uniamo delle piste della scheda che altrimenti non potevano essere collegate.





Impareremo ad utilizzare l'altoparlante della scheda di ingressi e uscite, a questo scopo realizzeremo un programma, l'esempio 6, con possibilità di attivarlo e disattivarlo in modo periodico. L'altoparlante è collegato al piedino RB7 del microcontroller PIC. Per attivarlo dobbiamo chiudere il jumper JMP3, e collegarlo alla morsettiera, facendo attenzione alla polarità.

Intestazione del programma. Come sempre, sceglieremo per prima cosa il modello di microcontroller e daremo un'etichetta ai registri che intendiamo utilizzare durante il programma. Infine collocheremo i vettori 0 e 5 per posizionare la prima istruzione del programma, e per saltare il vector di interrupt.

```
c:\progra\Amplab\esempio6.asm
1  ;*****
2  ;Programma contatore
3
4      list    p=16F84      ;Tipo di dispositivo
5
6  ;Area delle etichette
7  STATA EQU    0x03      ;Registro di stato
8  PORTA EQU    0x05      ;Porta A
9  PORTB EQU    0x06      ;Porta B
10 RPO EQU    0x05
11 OPTION_REG EQU 0x01      ;Timer0 e Option
12 INTCON EQU    0x0B
13 ;Registri di utilizzo generale
14 TEMP01 EQU    0x0D
15
16      ORG    0      ;Vector Reset
17
18      goto  INIZIO
19      ORG    5      ;Saltiamo il vector di
20                      ;interrupt.
```

```
c:\progra\Amplab\esempio6.asm
21
22 ;Inizio del programma
23
24 INIZIO: bsf    STATUS, 7
25         movlw b'01111111'
26         movwf PORTB
27         movlw b'00001111'
28         movwf OPTION_REG
29         bcf    PORTB
30         clrf   PORTB
31 ;Ciclo principale.
32
33 CICLO: bsf    PORTB, 7
34         call  DELAY1S
35         bcf    PORTB, 7
36         call  DELAYMS
37         goto  CICLO
38
39
40
```

La configurazione è semplice. RB7, dove si trova l'altoparlante, deve essere un'uscita; inoltre assegneremo il prescaler al Timer0, dato che realizzeremo delle temporizzazioni. Il ciclo principale del programma consiste nell'attivare e disattivare il segnale del piedino RB7, realizzando due temporizzazioni diverse, una durante l'accensione e l'altra durante lo spegnimento

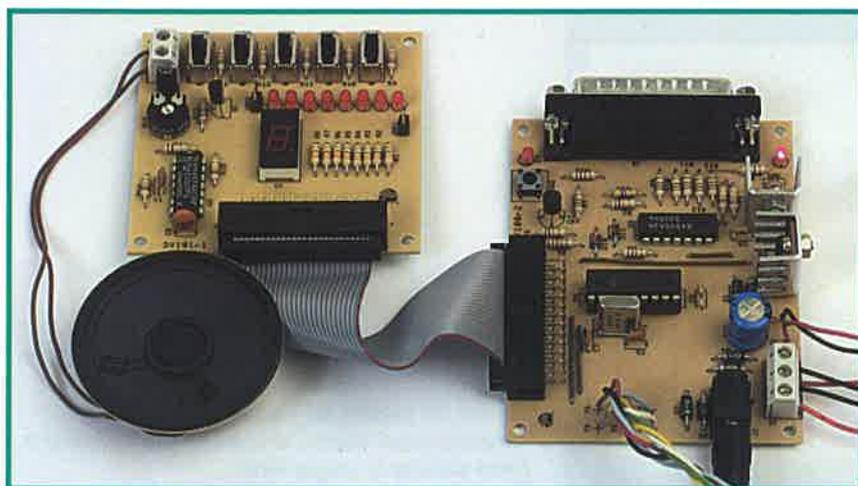
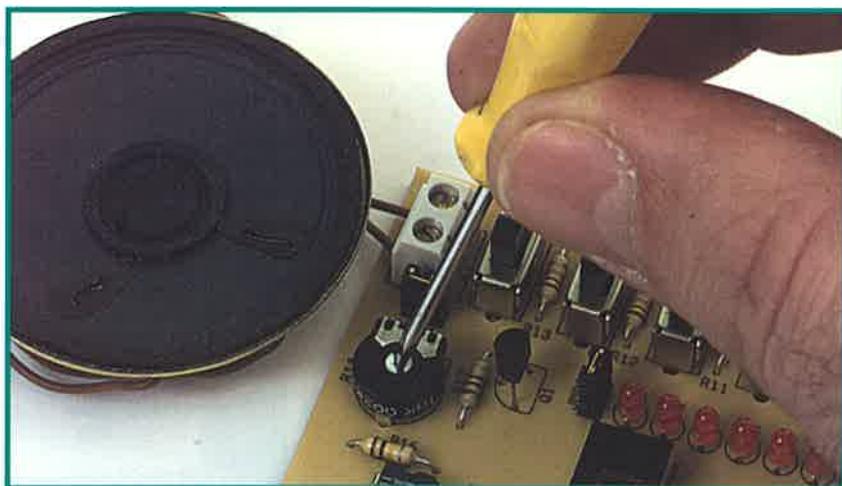
```

c:\Aprogra\T\Amplab\esempiob.asm
40
41 ;Routine di ritardo
42
43 DELAY1S bcf     STATUS, RP0
44         movlw  .100
45         movwf  TEMP01
46         call  DEL10
47         return
48
49 DELAYMS bcf     STATUS, RP0
50         movlw  .50
51         movwf  TEMP01
52         call  DEL10
53         return
54
55 DEL10   bcf     INTCON,2
56         movlw  0x08
57         movwf  OPTION_REG
58
59 DEL10_1 btfss  INTCON, 2
60         goto  DEL10_1
61         decfsz TEMP01, 1
62         goto  DEL10
63         return
64
65         END    ;Fine del programma
66 :*****

```

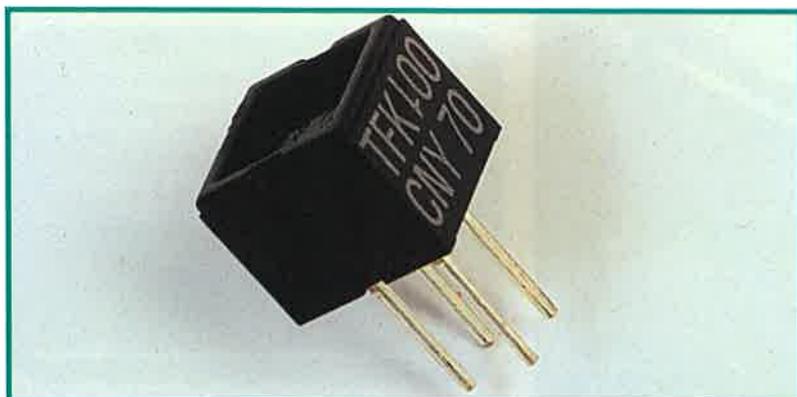
Queste sono le due routine di temporizzazione impiegate. La prima di queste, DELAY1S la conosciamo già, e realizza una temporizzazione di 1 secondo. La routine DELAYMS temporizza mezzo secondo. Le due routine si appoggiano alla routine DELAY10, che temporizza 10 millisecondi. Nell'altoparlante si otterrà un suono simile a quello di una sveglia.

Potremo regolare il timbro o il tono dell'altoparlante muovendo con un cacciavite il potenziometro R18. Questo fa sì che il generatore di onda quadra della scheda di ingressi e uscite vari la frequenza che si applica all'altoparlante. In questo modo possiamo scegliere il suono che più ci piace.



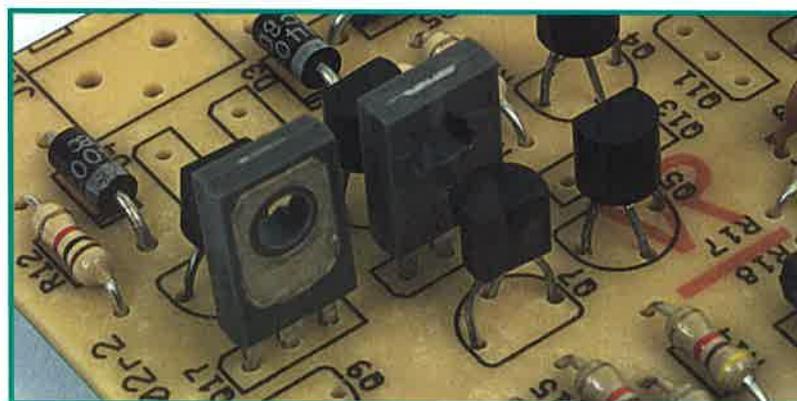
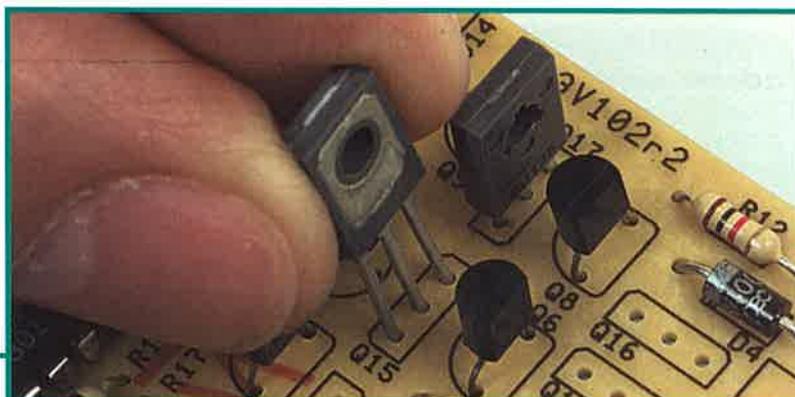
Come già sappiamo fare, dopo aver compilato il programma lo memorizzeremo sul microcontroller, mediante la scheda di controllo. Fate attenzione alla corretta alimentazione della scheda, così come alla posizione dell'interruttore di scrittura. Durante la scrittura il connettore Picbus deve essere collegato. Dopo aver memorizzato il programma, raccomandiamo di togliere il cavo DB25 dalla scheda di controllo.

Montaggio passo a passo



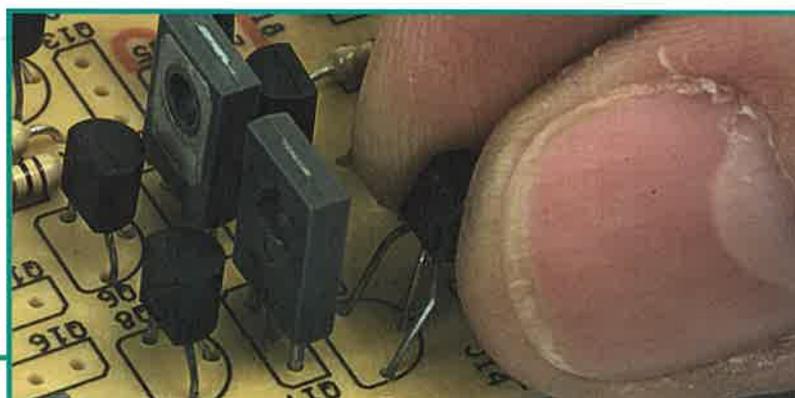
Questo è un sensore ottico a riflessione, modello CNY70. Servirà per controllare il movimento e la posizione di Monty. Il suo segnale arriverà alla scheda di controllo, sul quale si trova il microcontroller, attraverso la scheda di potenza. Più avanti spiegheremo come fare le opportune connessioni.

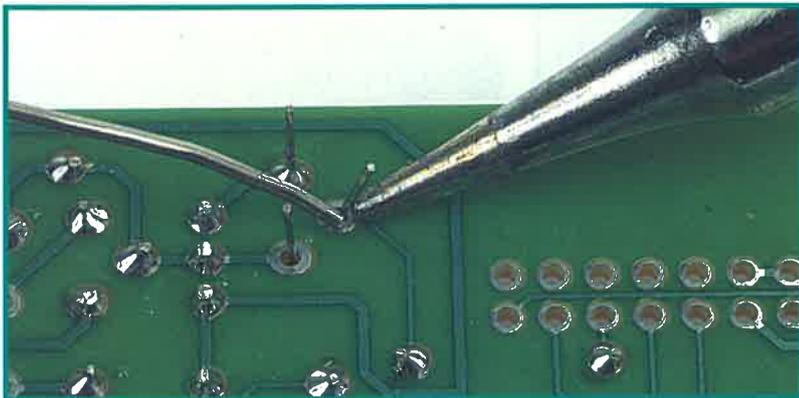
Montiamo ora il transistor modello BD135. Dobbiamo inserirlo nella posizione Q15, a lato dell'altro transistor BD135 che abbiamo già montato e saldato.



Inseriremo il transistor nel verso mostrato nell'immagine. Osservate che il transistor non è inserito totalmente nella scheda, ma rimane sollevato di qualche millimetro sulla superficie della medesima. Per ragioni estetiche tutti i transistor devono rimanere alla stessa altezza.

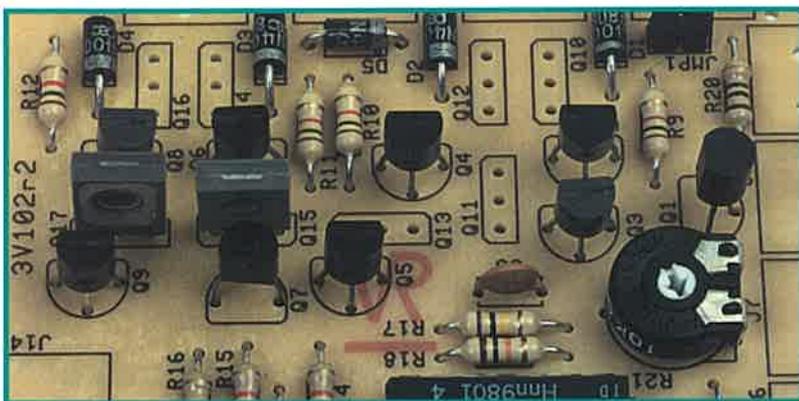
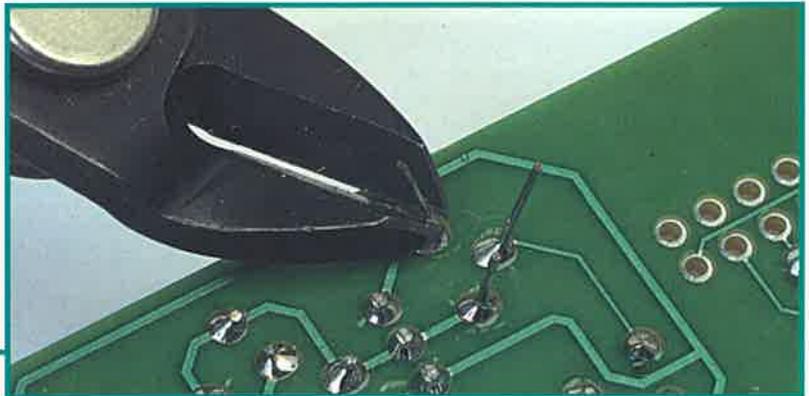
Ora collegheremo l'altro transistor NPN modello BC547. Verrà inserito in Q9, nel verso mostrato dall'immagine. Anche in questo caso non dovremo inserire il componente sino in fondo, ma lasciarlo sollevato dalla scheda qualche millimetro.





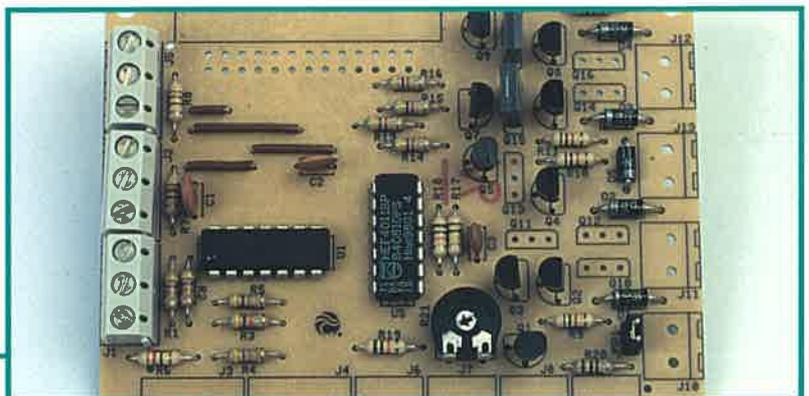
Salderemo il transistor, facendo attenzione a mantenere diritti i piedini il più possibile, in modo da ottenere un buon cono di stagno dopo aver realizzato la saldatura. Dobbiamo riempire di stagno tutta la superficie della piazzola di saldatura di ogni piedino.

Dopo aver saldato il componente procederemo a tagliare la parte di reoforo in eccesso. Non dobbiamo tirare il piedino durante il taglio, perché si potrebbero staccare le piste. Il taglio deve essere netto.



Dopo aver saldato il transistor BC547, avremo saldato tutti i transistor di questo modello sulla scheda di potenza, che vanno da Q1 sino a Q9. Dobbiamo verificare di averli saldati tutti nel verso mostrato dall'immagine.

Questo è l'aspetto che deve avere la scheda di potenza con tutti i componenti che abbiamo saldato sino a questo momento. Dobbiamo verificare il corretto inserimento di ognuno di essi.



```

c:\progra~1\mplab\esempio7.asm
1      list p=16F84          ;Tipo di dispositivo
2
3      ;Area delle etichette
4      STATUS EQU 0x03      ;Registro di stato
5      PORTA EQU 0x05      ;Porta A
6      PORTB EQU 0x06      ;Porta B
7      RP0 EQU 0x05
8      OPTION_REG EQU 0x01 ;Timer0 e Option
9      INTCON EQU 0x0b
10
11     EECON1 EQU 88h      ;Etichetta di registro
12     EECON2 EQU 89h      ;per la EEPROM
13     EEDATA EQU 88h
14     EEADR EQU 09h
15
16     RP0 EQU 5          ;Etichetta di bit
17     EEIE EQU 6
18     WREN EQU 2
19     WR EQU 1
20     RD EQU 0
21
22     ;Registri di utilizzo generale
23     TEMP01 EQU 0x0D
24
25     ORG 0              ;Vector Reset
26
27     goto INIZIO
28     ORG 5              ;Saltiamo il vector di
29     ;Interrupt.
30

```

In questo esempio impareremo a scrivere i dati nella memoria EEPROM interna del microcontroller. I dati che vengono scritti in questa memoria non sono volatili come quelli della memoria RAM, e rimangono inalterati anche quando togliamo l'alimentazione al PIC. Quindi è una memoria ideale in cui scrivere chiavi o dati che devono rimanere inalterati. All'inizio del programma definiamo come al solito il tipo di dispositivo e le etichette dei registri da impiegare.

Il programma scriverà all'indirizzo zero della memoria EEPROM il dato corrispondente al valore presente nella porta A, il quale sarà selezionato dagli interruttori della scheda di ingressi e uscite. A questo scopo configureremo la porta A come ingresso, chiameremo la routine DATO, che raccoglie il dato presente sulla porta A, e la routine WRITE che lo scrive nella memoria EEPROM.

```

c:\progra~1\mplab\esempio7.asm
30
31 ;Inizio del programma
32
33 INIZIO bsf STATUS, RP0
34        movlw b'00011111'
35        movwf PORTA
36        movlw b'00000111'
37        movwf OPTION_REG
38        bcf STATUS, RP0
39        call DATO
40        call WRITE
41        goto FINE

```

```

c:\progra~1\mplab\esempio7.asm
45
46 DATO   bcf STATUS, RP0
47        movlw b'00000000'
48        movwf EEADR
49        movf PORTA, W
50        movwf EEDATA
51        return
52
53 WRITE  bsf STATUS, RP0
54        bsf EECON1, WREN
55        movlw 0x55
56        movwf EECON2
57        movlw 0xAA
58        movwf EECON2
59        bsf EECON1, WR
60        bcf STATUS, RP0
61        call DEL10
62        return

```

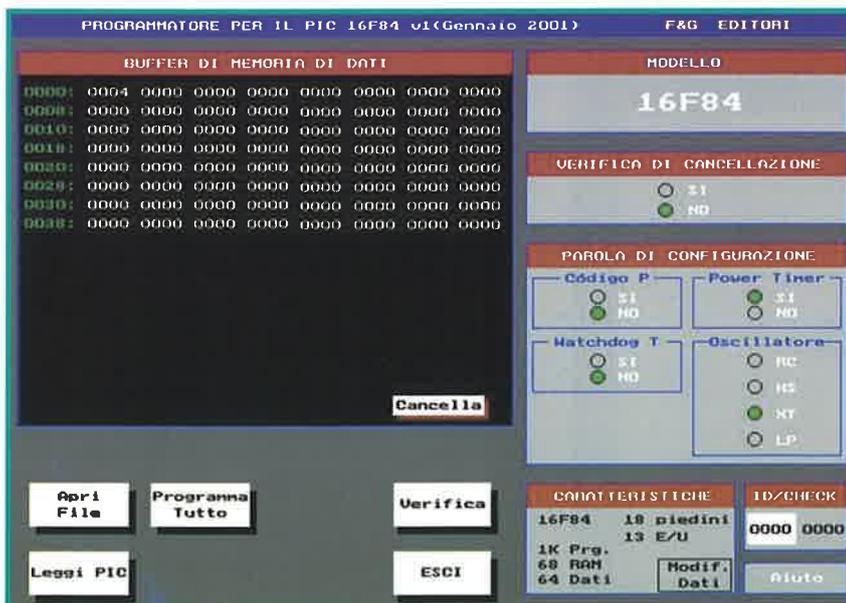
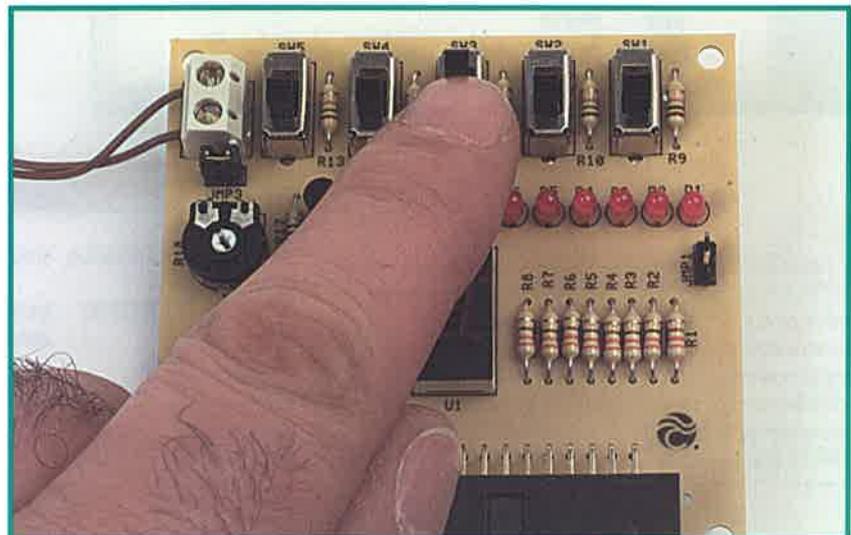
La routine DATO seleziona l'indirizzo di memoria EEPROM da scrivere, in questo caso l'indirizzo zero, e legge il dato della porta A, scrivendolo in EEDATA, registro che deve contenere il valore che si vuole immagazzinare nella memoria EEPROM. La routine WRITE realizza il processo necessario per scrivere i dati nella memoria EEPROM, utilizzando i registri EECON1 ed EECON2. Si abilita il permesso di scrittura (WREN), in EECON2 si inserisce una sequenza stabilita da Microchip, infine si inizia il ciclo di scrittura, attivando WR.

```

c:\progra~1\mplab\esempio7.asm
63
64
65 ;Routine di ritardo
66
67 DEL10   bcf     INTCON, 2
68         movlw  0xD8
69         movwf  OPTION_REG
70
71 DEL10_1 btfsz  INTCON, 2
72         goto  DEL10_1
73         decfsz TEMP01, 1
74         goto  DEL10
75         return
76
77         END    ;Fine del programma
78 ;*****
    
```

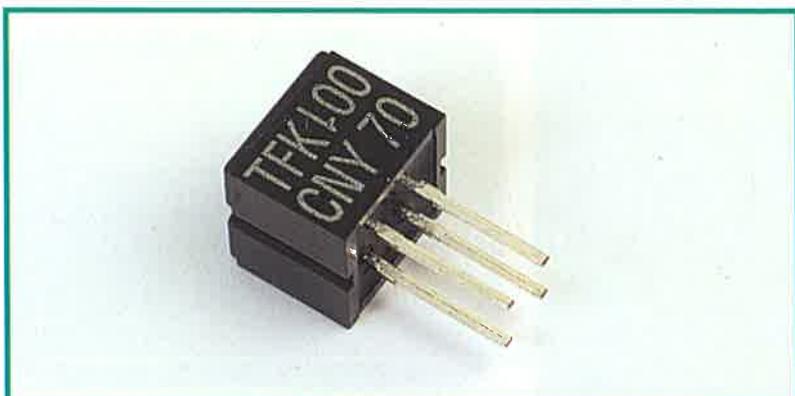
Questa routine di temporizzazione è necessaria, dato che il processo di scrittura di un dato nella memoria EEPROM non è così immediato come succede per la memoria RAM, ma può impiegare sino a 10 ms per compiersi. Anche se per questo programma potremmo fare a meno della temporizzazione, visto che scriviamo un solo dato, dobbiamo abituarci a considerare questa necessità. Un altro modo di verificare se è terminata la scrittura, consiste nel testare il bit EEIF del registro EECON1, che viene messo a 1 alla fine di detto ciclo.

Terminato il programma, selezioneremo una combinazione con gli interruttori della scheda di ingressi e uscite, e premeremo il pulsante di reset del microcontroller. Dopo il reset, il PIC esegue il programma e scrive all'indirizzo zero della memoria EEPROM il dato scritto con gli interruttori.



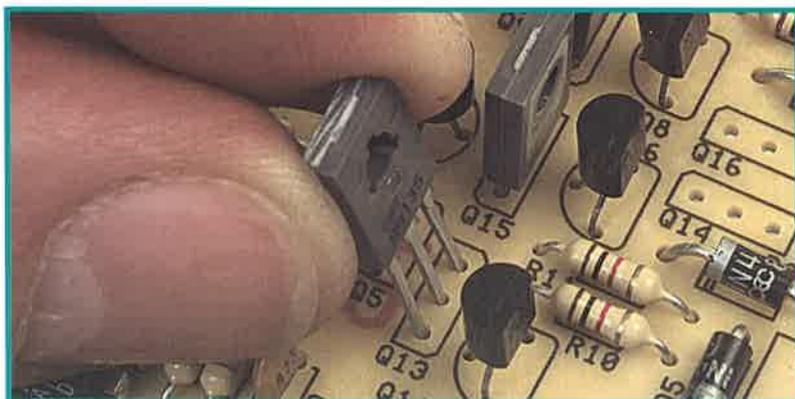
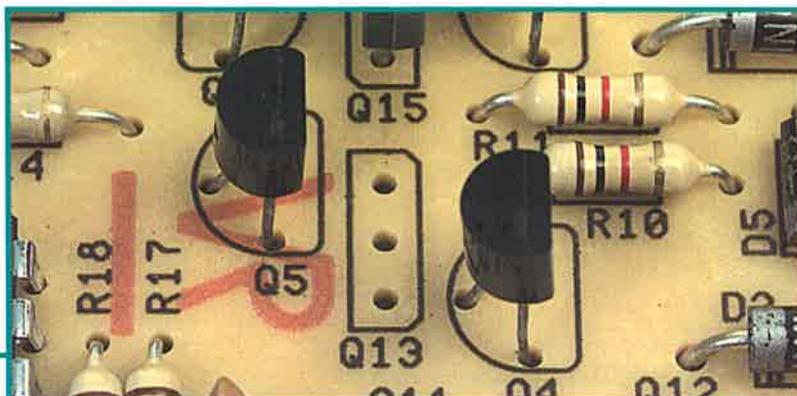
Per verificare di aver scritto bene il dato nella memoria EEPROM, possiamo eseguire il programma di scrittura e selezionare l'opzione Modifica Dati. Nella nuova videata che appare sceglieremo Leggi PIC. Sul video apparirà all'indirizzo zero, il valore del dato memorizzato.

Montaggio passo a passo



Ora disponiamo di un altro sensore CNY70, dei tre che equipaggeranno Monty. Con due di questi sensori controlleremo perfettamente la direzione del robot, e con l'altro la velocità e la distanza percorsa. Più avanti vi spiegheremo come realizzare i collegamenti con questo tipo di sensori.

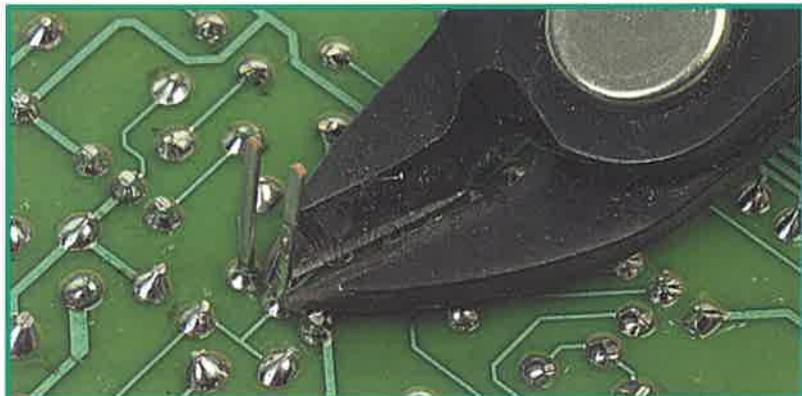
Monteremo e salderemo l'altro transistor BD135, inserendolo in Q13. Come possiamo vedere sulla serigrafia della scheda, uno dei lati lunghi è chiuso da due angoli retti, mentre l'altro lato ha gli angoli smussati. La parte dritta corrisponde alla parte anteriore del transistor, su cui troviamo una zona metallica.



Seguendo le indicazioni fornite in precedenza, monteremo il transistor così come mostrato nell'immagine: sollevato di qualche millimetro rispetto alla scheda.

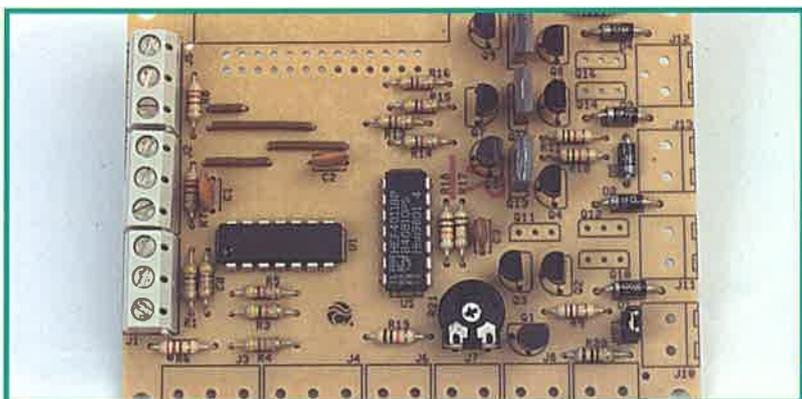
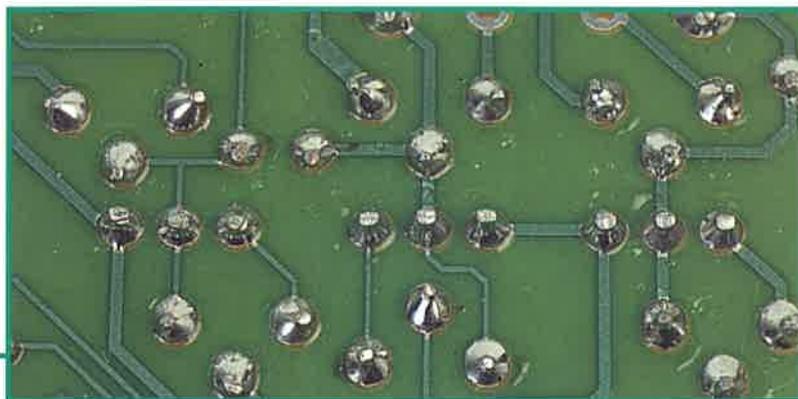
Dopo esserci assicurati della corretta inserzione e dell'orientamento del transistor, procederemo alla saldatura. Un componente già saldato, è molto difficile da dissaldare, inoltre si corre il rischio di rovinare la scheda.





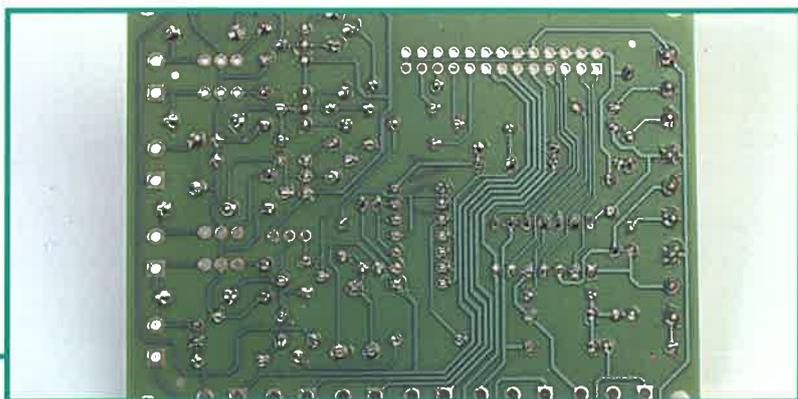
Dopo aver realizzato le saldature, che devono avere un aspetto brillante e forma conica taglieremo la parte di reoforo in eccesso. Raccomandiamo di utilizzare un tronchesino simile a quello della figura.

Questo è l'aspetto che hanno le diverse saldature dei transistor già realizzate. Osserviamo che le saldature dei transistor BD135 hanno un po' più di stagno rispetto a quelle dei BC547. Tutte formano un cono, che riempie completamente la piazzola da saldare.



Aspetto della scheda di potenza, con tutti i componenti forniti sino ad ora, montati e saldati. A questo punto la maggior parte dei componenti discreti sono già stati montati.

Questa immagine mostra come devono essere realizzate tutte le saldature fatte fino a questo momento. Consigliamo di verificare che tutte siano ben appoggiate sulle piazzole e che non formino delle bolle sui piedini.



```

c:\progra~1\mplab\esempio8.asm
1  ;*****
2  ;Programma contatore
3
4      list p=16F04          ;Tipo di dispositivo
5
6  ;Area delle etichette
7  STATUS EQU 0x03          ;Registro di stato
8  PORTA EQU 0x05          ;Porta A
9  PORTB EQU 0x06          ;Porta B
10 RP0 EQU 0x05
11 OPTION_REG EQU 0x01      ;Timer0 e Option
12 INTCON EQU 0x0B
13
14 EECON1 EQU 08h          ;Etichetta di registro
15 EEDATA EQU 08h
16 EEADR EQU 09h
17
18 RP0 EQU 5              ;Etichetta di bit
19 RD EQU 0
20
21 ;Registri di utilizzo generale
22 TEMPO1 EQU 0x0D
23
24 ORG 0                  ;Vector Reset
25
26 goto INIZIO
27 ORG 5                  ;Saltiamo il vector
28 ;di interrupt.
29

```

Abbiamo già visto come scrivere i dati nella memoria EEPROM e verificare di averli scritti correttamente tramite il software di scrittura. Nell'esempio 8, che ora vi proponiamo, impareremo a leggere i dati dalla memoria EEPROM per utilizzarli nel nostro programma. La prima cosa che dobbiamo fare nel programma è, come al solito, scegliere il dispositivo e posizionare le etichette pertinenti.

Il programma inizia all'indirizzo 5. Leggeremo il dato che abbiamo scritto all'indirizzo 0 della memoria EEPROM (con il programma del precedente esempio 7) e lo visualizzeremo sulla fila dei LED. Per poter fare questo, configuriamo la porta B come uscita, chiamiamo la routine READ, e posizioniamo il dato letto sulla porta B.

```

c:\progra~1\mplab\esempio8.asm
29
30 ;Inizio del programma
31
32 INIZIO: bsf STATUS, RP0
33         clrfs PORTB
34         bcf STATUS, RP0
35         clrfs EEADR
36         call READ
37         movf EEDATA, W
38         movwf PORTB
39
40 FINE:   goto FINE
41
42

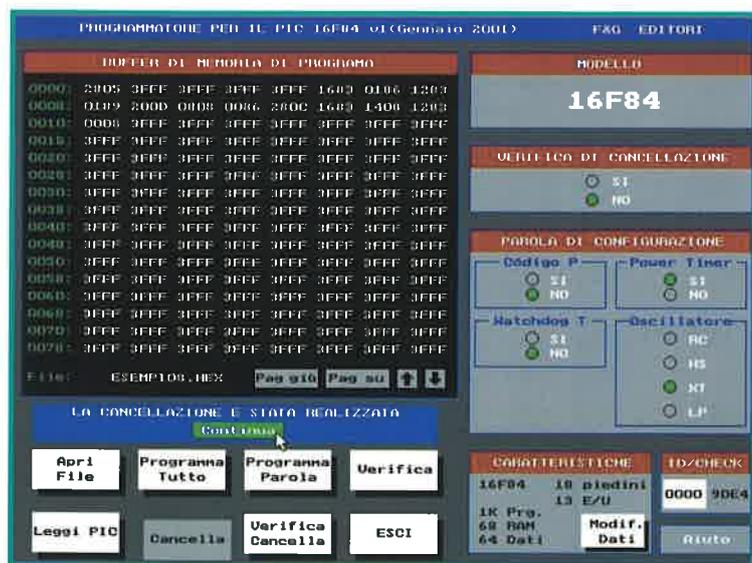
```

```

c:\progra~1\mplab\esempio8.asm
42
43 READ   bsf STATUS, RP0
44        bsf EECON1, RD
45        bcf STATUS, RP0
46        return
47
48        END ;Fine del programma
49 ;*****

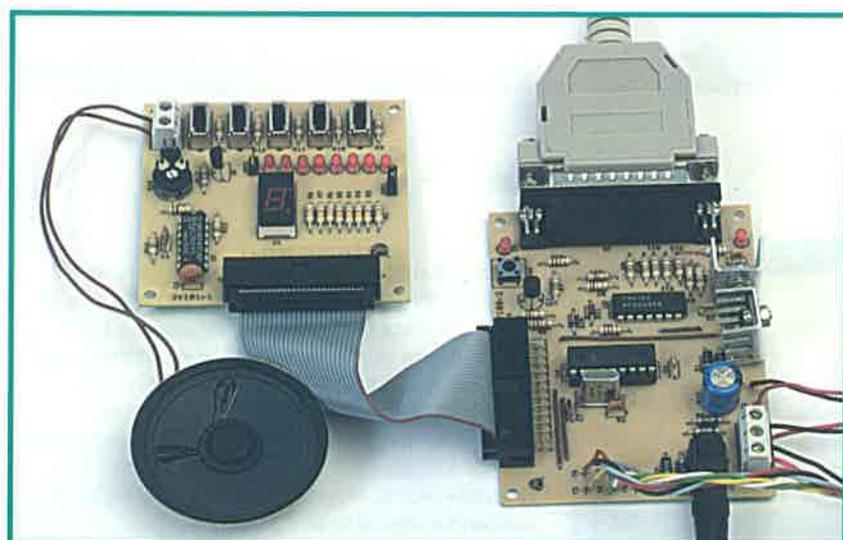
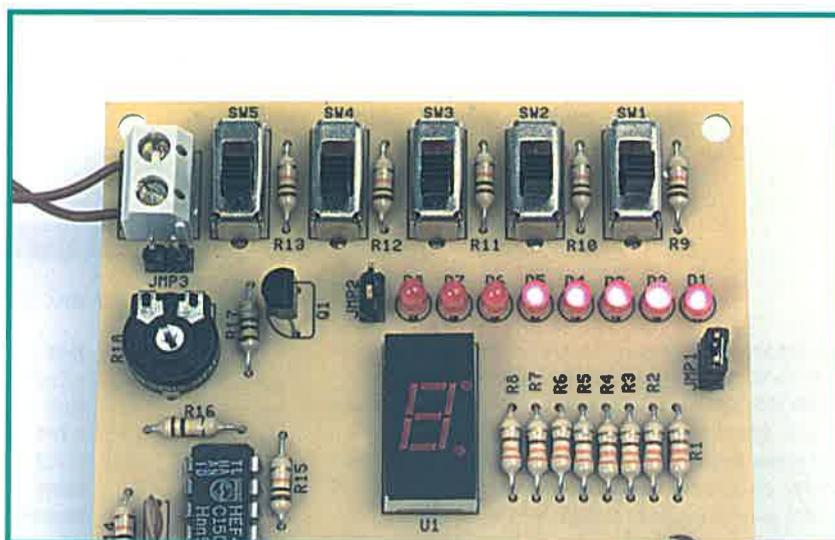
```

Come si può vedere, la routine per leggere la memoria EEPROM dei dati è molto più semplice di quella utilizzata per scrivere. A differenza della scrittura, la lettura dei dati non comporta una significativa perdita di tempo, inoltre possiamo leggere tanti dati di seguito quanti vogliamo. Carichiamo nel registro EEADR l'indirizzo desiderato, tramite il registro EECON1 attiviamo il ciclo di lettura (RD) e, finalmente, nel registro EEDATA abbiamo a disposizione il dato letto.



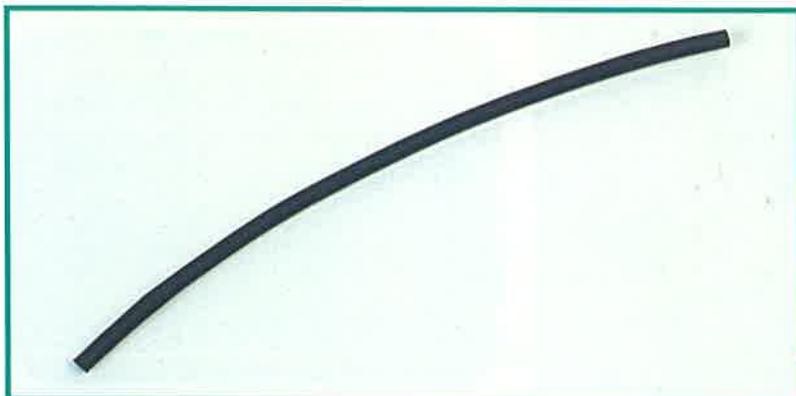
Dopo aver compilato il programma procederemo alla sua scrittura nel microcontroller. Prima di eseguire la scrittura cancelleremo la memoria di programma, come facciamo sempre. Cancellare la memoria di programma del PIC non significa cancellare la memoria dei dati del medesimo, per questo i dati che sono memorizzati nella EEPROM rimangono inalterati anche dopo la cancellazione del PIC.

Per vedere il risultato sui LED, dobbiamo chiudere il jumper JMP1 della scheda di ingressi e uscite, e lasciare gli altri jumpers aperti. Il dato che possiamo osservare nell'immagine corrisponde fedelmente al dato scritto tramite la porta A, con il programma precedente.



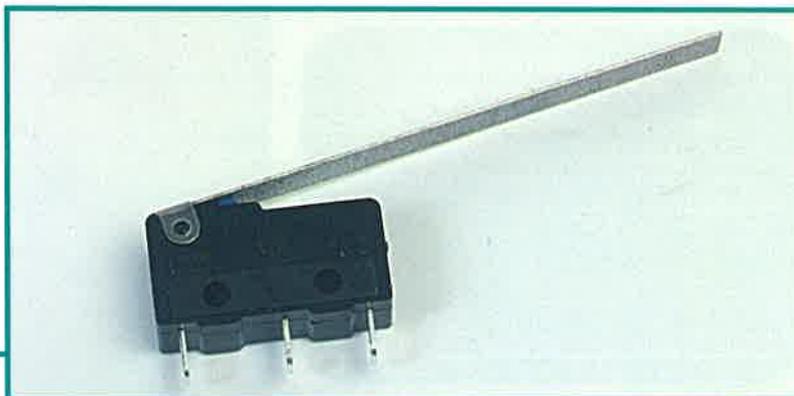
La fotografia ci mostra i collegamenti necessari per provare il programma. È necessario alimentare il sistema attraverso la scheda di controllo, programmare il PIC, e una volta scritto collegare, tramite il Picbus, la scheda di ingressi e uscite. Raccomandiamo di scollegare il connettore DB25 dopo aver programmato il microcontroller.

Montaggio passo a passo



Tubo termorestringente da 3 mm di diametro, che utilizzeremo per isolare i piedini dei sensori dopo averli montati e saldati ai cavi. Grazie al termorestringente eviteremo che i piedini dei sensori entrino in contatto fra di loro.

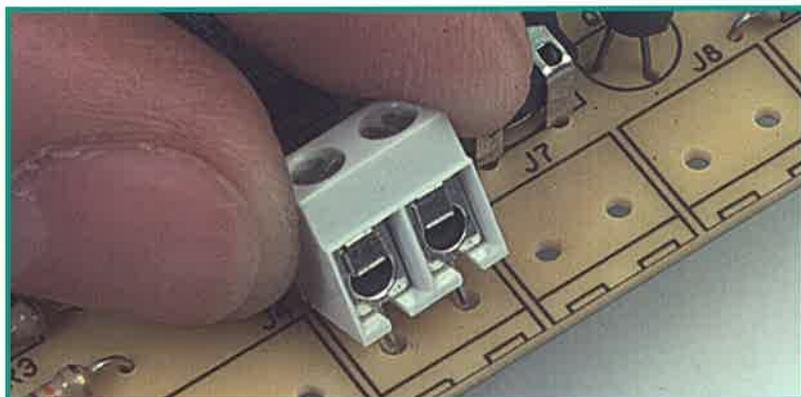
Sensore a levetta, o finecorsa. Si tratta di un sensore di tipo meccanico, come un interruttore, che servirà a Monty per rilevare gli ostacoli, oppure per evitare un possibile scontro. Sarà collegato tramite la scheda di potenza.



Salderemo ora il transistor modello BD135. Verrà montato in Q11, con il verso di inserzione mostrato dall'immagine, e che corrisponde alle indicazioni sulla serigrafia che conosciamo già.

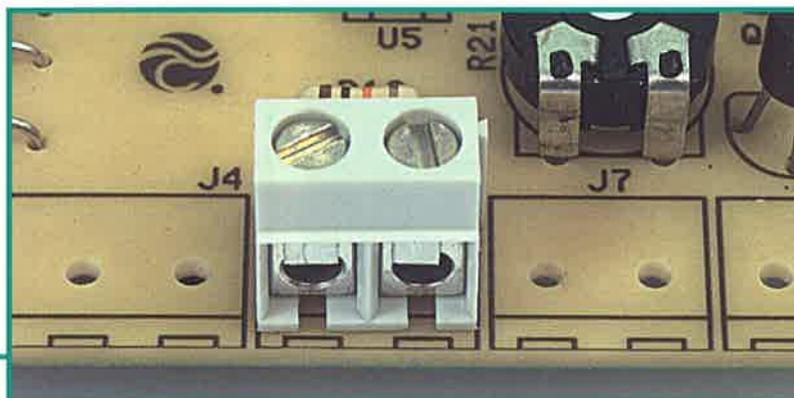
Dopo aver inserito il transistor, e averlo lasciato sollevato di alcuni millimetri dalla superficie della scheda, lo salderemo come sappiamo già fare. Dopo averlo saldato taglieremo la parte dei reofori in eccesso.



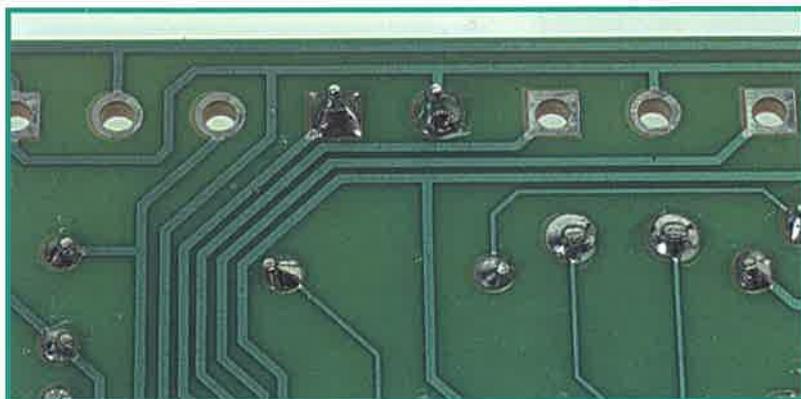


Ora monteremo una morsettiera a due contatti, che sarà inserita in J6, con gli ingressi orientati verso l'esterno della scheda, come mostrato dall'immagine.

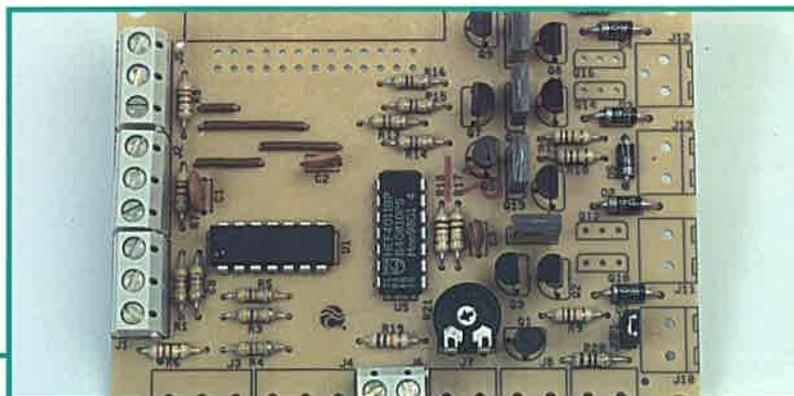
Prima di eseguire le saldature dobbiamo assicurarci che la morsettiera sia inserita completamente a filo della scheda. Raccomandiamo di saldare prima un solo piedino, e verificare nuovamente il corretto verso di inserzione.

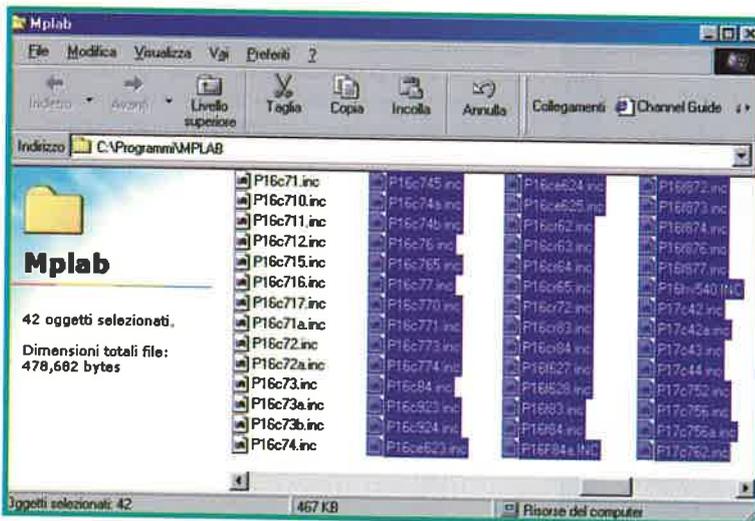


Le saldature per le morsettiere richiedono più stagno del solito. Riempiremo di stagno tutta la piazzola di saldatura, sino a formare un cono.



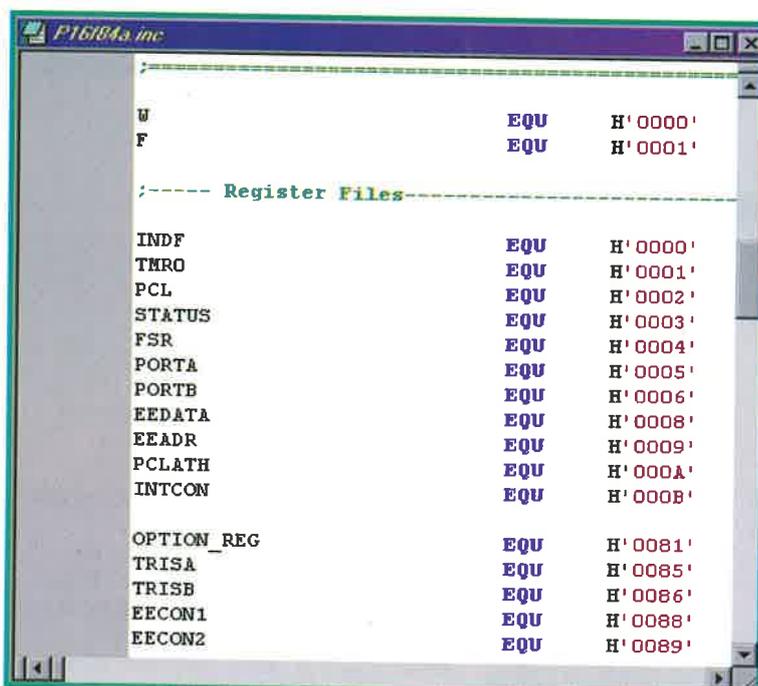
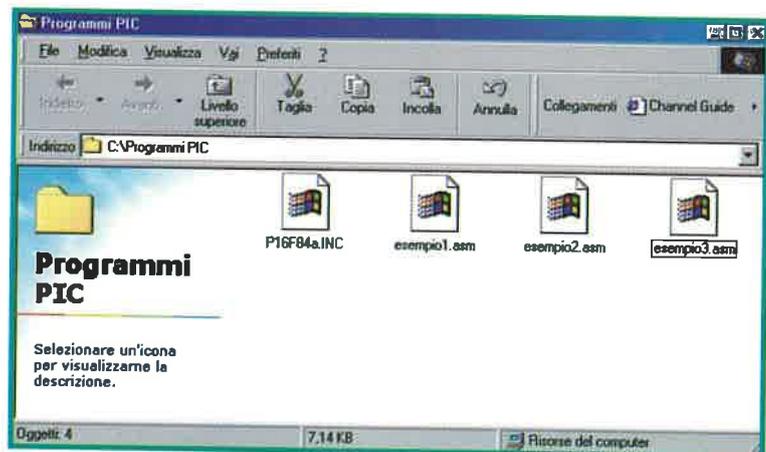
Vista dall'alto della scheda di potenza con tutti i componenti saldati sino a questo momento. Rimangono da saldare quattro transistor e il resto dei connettori e delle morsettiere.





Impareremo a includere i file nel nostro programma sorgente. Uno dei file più utili che dobbiamo includere, e che utilizzeremo sempre a partire da questo momento, è quello che contiene tutte le definizioni dei registri interni del microcontroller. Questi file si trovano nella cartella dove abbiamo installato MPLAB.

Utilizzeremo il file P16F84A.INC, che è quello che possiede la definizione dei registri del nostro microcontroller. Dobbiamo copiare questo file nella stessa directory in cui si trovano i programmi sorgente, in cui vogliamo includere questo file.



Se apriamo e editiamo il file P16F84A.INC, possiamo verificare quali sono i nomi che sono stati assegnati ad ognuno dei registri del microcontroller. Sono gli stessi nomi che abbiamo utilizzato sino ad ora, e che utilizza ufficialmente Microchip sulla documentazione tecnica, per denominare i registri.

```

P16F84a.inc
;----- STATUS Bits -----
IRP                EQU    H'0007'
RP1                EQU    H'0006'
RP0                EQU    H'0005'
NOT_TO             EQU    H'0004'
NOT_PD             EQU    H'0003'
Z                  EQU    H'0002'
DC                 EQU    H'0001'
C                  EQU    H'0000'

;----- INTCON Bits -----
GIE                EQU    H'0007'
EEIE               EQU    H'0006'
TOIE               EQU    H'0005'
INTE               EQU    H'0004'
RBIE               EQU    H'0003'
TOIF               EQU    H'0002'
INTF               EQU    H'0001'
RBIF               EQU    H'0000'

;----- OPTION_REG Bits -----

```

Oltre ai registri, in questo file si trovano anche tutti i bit di quei registri che hanno qualche significato speciale. Dobbiamo familiarizzare con questi nomi, dato che saranno molto più semplici da ricordare rispetto alla loro posizioni interne in ogni registro.

Per includere il file di dichiarazione dei registri P16F84A.INC, scriveremo nell'intestazione del programma: "include P16F84A.INC". In seguito potremo dichiarare i nomi che vogliamo ai nostri registri di lavoro, che si trovano a partire dall'indirizzo 0x0c.

```

c:\progra\1\mplab\esempio9.asm
1
2      LIST    p=16F84      ;Tipo di dispositivo
3
4      include "P16F84.INC" ;Definizione dei registri
5
6      ;Definizione dei registri di carattere generale
7
8      AUX      EQU    0x0c
9      CONTATORE EQU    0x0d
10     DATO     EQU    0x0e
11

```

```

c:\progra\1\mplab\esempio9.asm
13
14     #define   Pin_uscita   PORTB,0
15     #define   Pin_ingresso PORTA,1
16
17
18     ORG      INIZIO      0
19     goto
20
21     ORG      5
22
23     INIZIO:   bsf      STATUS, RP0
24             bcf      Pin_uscita
25             bsf      Pin_ingresso
26             bcf      STATUS, RP0
27
28     CICLO    goto     CICLO
29
30
31     END      ;Fine del programma

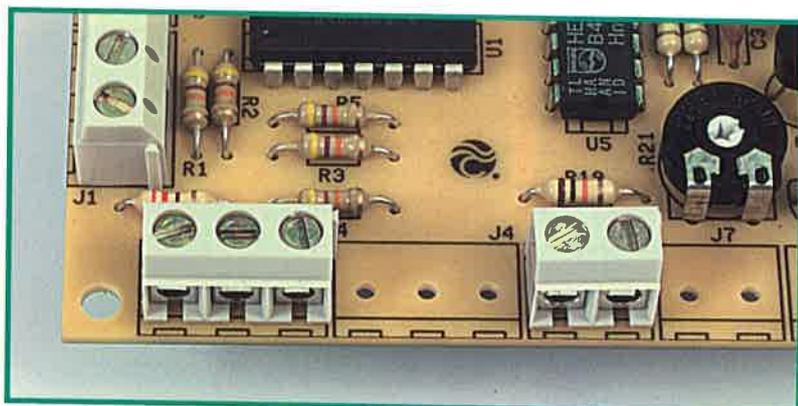
```

Oltre ai byte e ai bit, possiamo nominare anche i piedini reali delle porte. Come possiamo vedere nominiamo RBO come Pin_uscita e RA1 come Pin_ingresso, usando l'istruzione "define". Ora nel nostro programma faremo riferimento a questi nomi, senza la necessità di ricordare in che piedino si trovavano. Il presente esempio non è un programma utile, ma solo una dimostrazione di utilizzo delle definizioni.

MODULO DI POTENZA

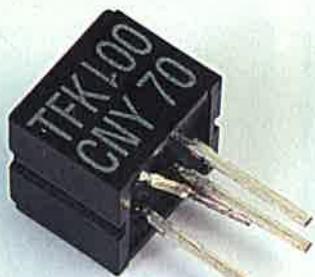
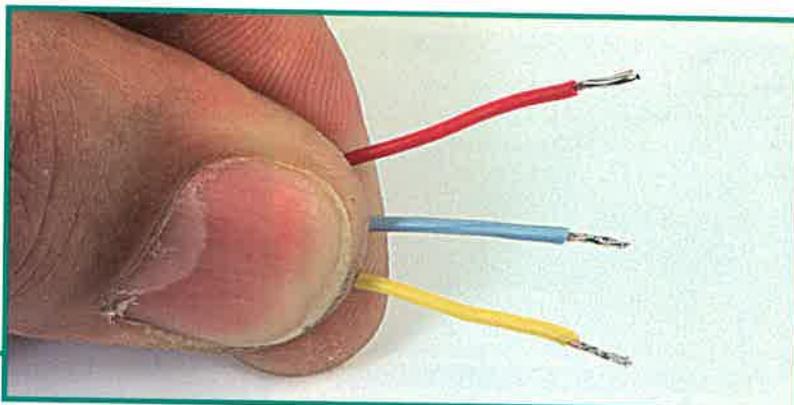
PL 58

Montaggio passo a passo



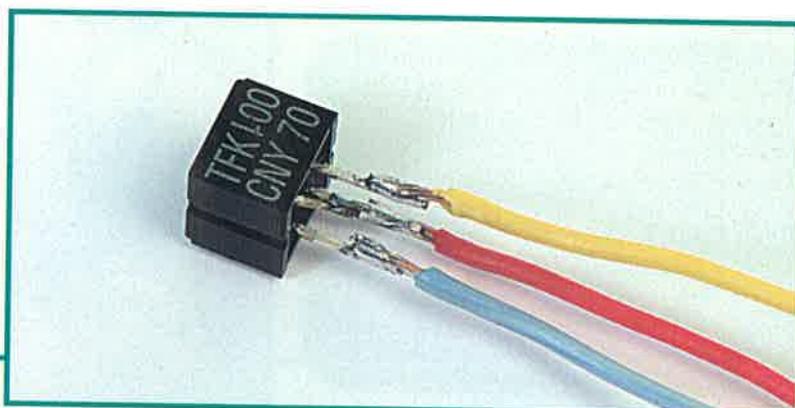
Il primo elemento da montare sarà la morsettiera a tre contatti J3. La inseriremo con le connessioni verso l'esterno della scheda, e ci assicureremo che si trovi totalmente appoggiata sulla superficie della stessa, prima di procedere alle saldature.

Ora monteremo i cavi ai sensori che già abbiamo. Prima taglieremo ognuno dei tre cavi, azzurro, rosso e giallo, in tre pezzi, dopo li speleremo agli estremi e li prestagneremo come mostrato nell'immagine.

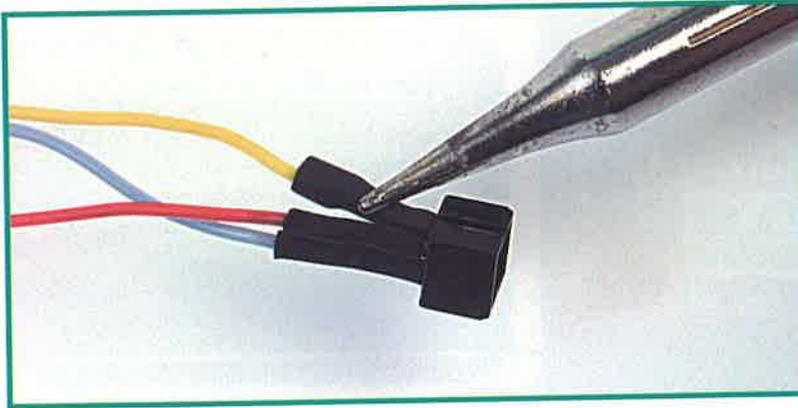


Il sensore CNY70 possiede quattro piedini, due dei quali devono essere uniti come mostrato nell'immagine. Una volta preparato il sensore, prestagneremo tutti i piedini.

Questo è l'aspetto che deve presentare il sensore CNY70 dopo aver saldato i tre cavi ai suoi piedini. Riscaldando con il saldatore il cavo prestagnato unito con il piedino del sensore, prestagnato anch'esso, deve formarsi la saldatura. Fate in modo che i colori dei cavi siano saldati ai piedini dei sensori secondo l'immagine.

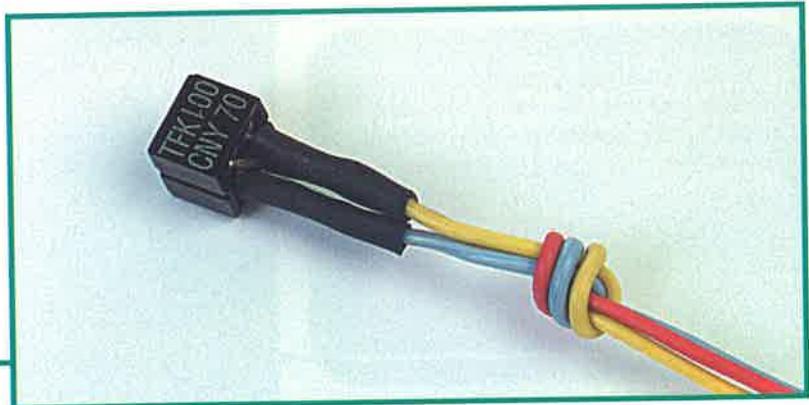


Montaggio passo a passo

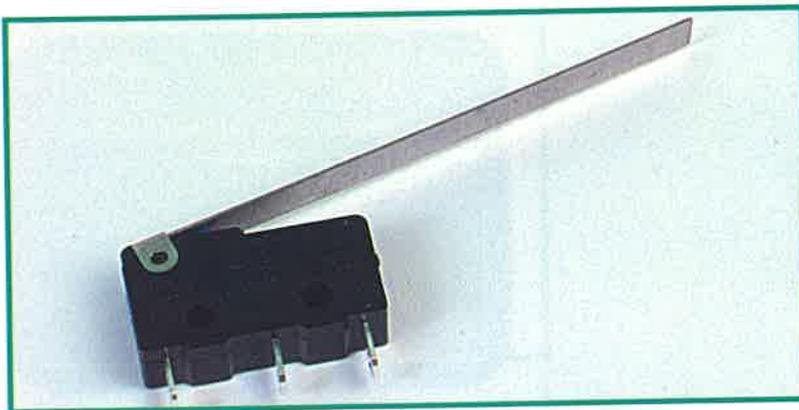


Una volta realizzate le saldature collegheremo un pezzo di tubo termorestringente in ognuno dei tre piedini. Dopo averlo montato lo riscaldiamo con il saldatore per farlo contrarre, affinché aderisca alle saldature. In questo modo non si produrranno cortocircuiti tra i piedini del sensore.

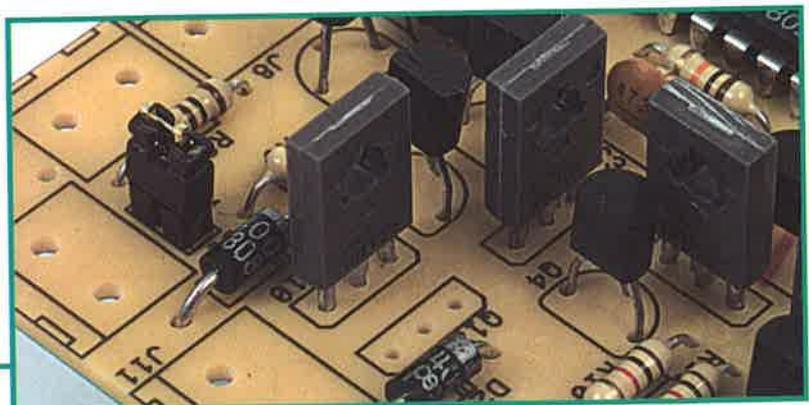
Aspetto finale che deve avere il sensore CNY70, pronto per essere collegato alla scheda di potenza. Ripetiamo lo stesso processo con gli altri sensori CNY70 che possediamo.

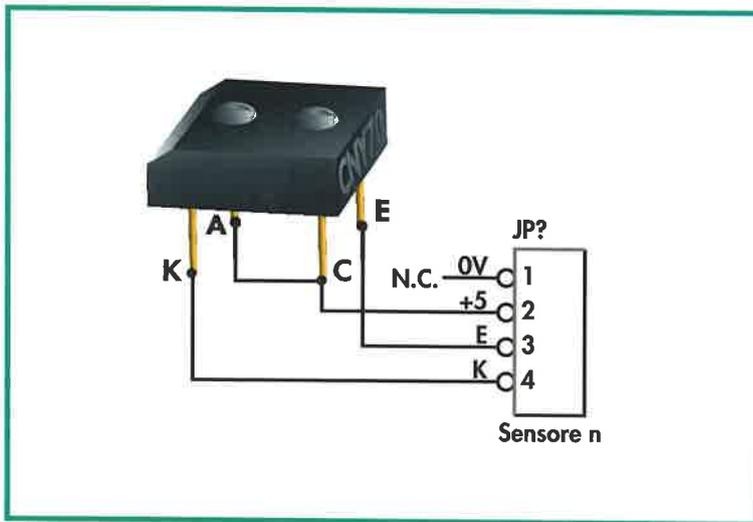


L'immagine mostra uno dei due finecorsa utilizzati da Monty, che prepareremo successivamente. I finecorsa sono anch'essi dei sensori, di tipo diverso. I sensori CNY70, preparati in precedenza, sono sensori di tipo ottico, non necessitano del contatto fisico con l'oggetto da rilevare. Al contrario, i finecorsa sono sensori di tipo elettromeccanico e necessitano del contatto fisico. Possono essere utilizzati per rilevare collisioni, oggetti, ecc.



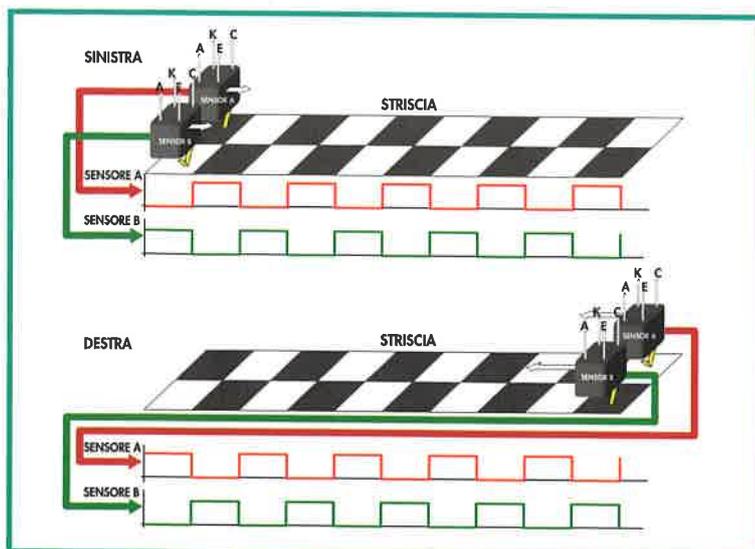
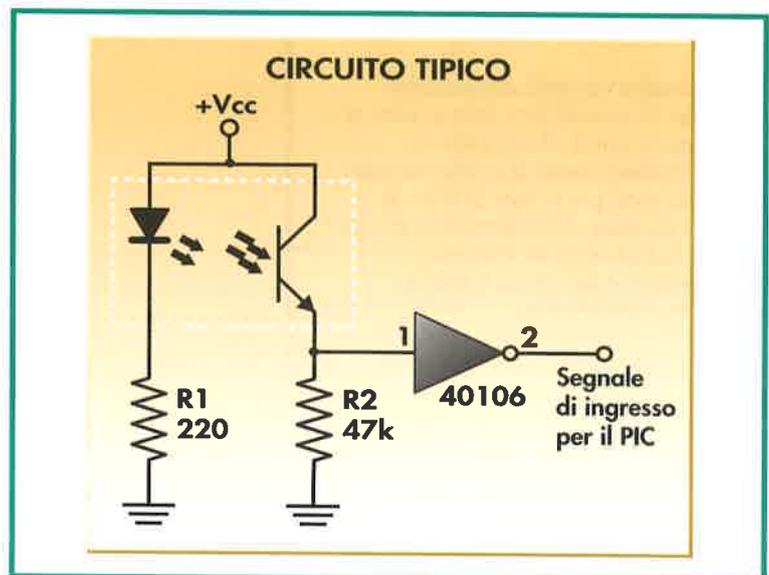
Infine salderemo sulla scheda di potenza un transistor BD135 che sarà inserito al suo posto, cioè in Q10. L'immagine mostra il verso corretto del montaggio.



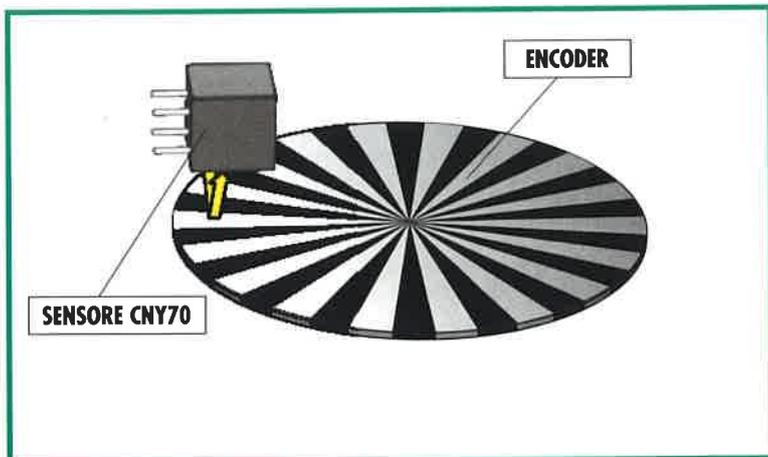


Questo è il collegamento che realizzeremo fra il sensore ottico CNY70 e la morsettiere di ingresso della scheda di potenza. Come possiamo osservare, realizzando le saldature dei cavi collegheremo insieme i due piedini indicati nella figura e li collegheremo a 5 V.

Circuito di interfaccia per il segnale del sensore, che si trova sulla scheda di potenza. È composto da due resistenze, una per il fotodiodo e l'altra per il fototransistor. Quando il raggio di luce che emette il fotodiodo colpisce la base del fototransistor, l'uscita del sensore produce uno 0, in caso contrario un 1. Questo è dovuto alla porta invertente 40106 che si trova alla fine del circuito, e permette di ottenere un buon segnale a livello TTL.

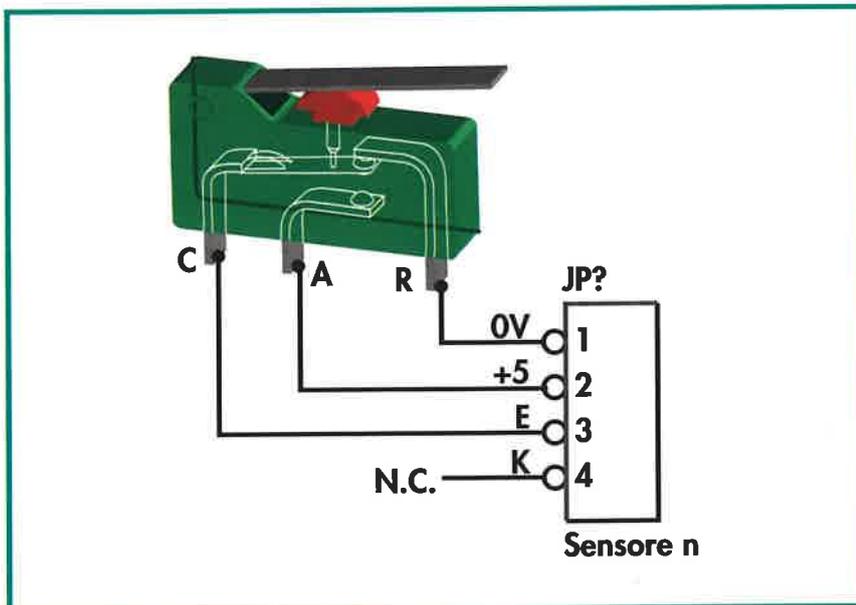
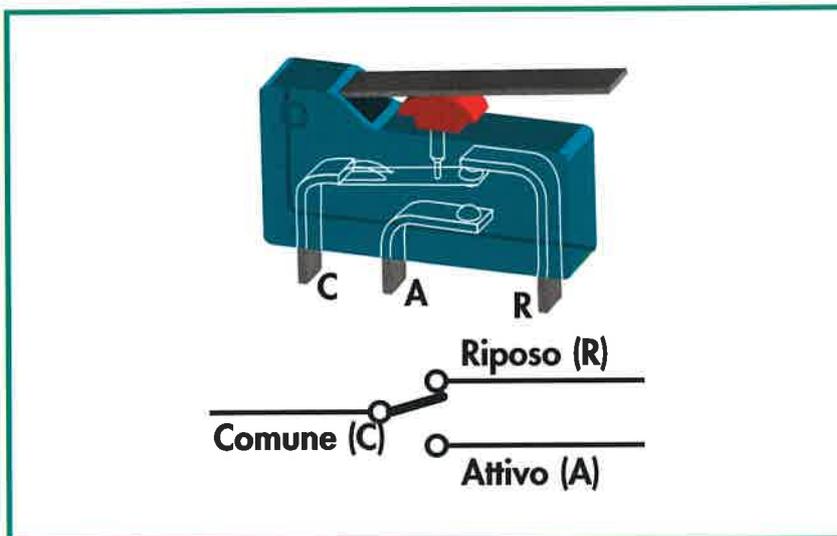


In questo cronogramma possiamo vedere il risultato dell'uscita dei due sensori CNY70 al passaggio su di un insieme di strisce nere e bianche. Dobbiamo familiarizzare con il funzionamento di questi sensori, perché saranno quelli che controlleranno la direzione del robot durante il suo movimento.



I sensori CNY70 possono anche essere utilizzati per il controllo della velocità del robot, oltre che per la sua direzione. Per fare questo possiamo montare un encoder associato ad una delle due ruote. Con un encoder dotato di strisce bianche e nere, si può misurare il tempo che impiega la ruota a compiere un giro, in questo modo si può calcolare la velocità di movimento del robot.

Schema interno del finecorsa o sensore a contatto. A seconda se si trova in stato di riposo oppure attivo, chiude uno dei suoi estremi rispetto al piedino centrale. Il finecorsa serve per rilevare gli ostacoli. Possiamo incollare, come supplemento, una prolunga sulla levetta del finecorsa, per aumentare il suo campo d'azione.

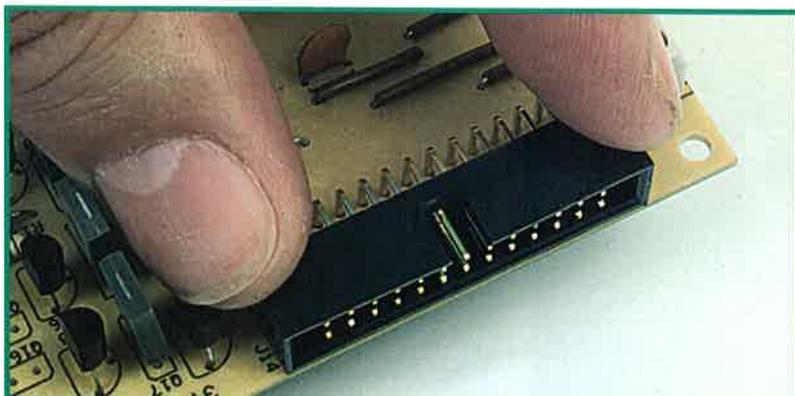


Questo è il modo di collegare il finecorsa alla scheda di potenza. Sui piedini destro e centrale collegheremo massa e 5 V, in modo da ottenere sul piedino di sinistra questi due livelli, a seconda che il sensore sia attivo oppure no. Inoltre mediante l'utilizzo di ultrasuoni, potremo rilevare la presenza di ostacoli senza la necessità di dover entrare in collisione con essi.

MODULO DI POTENZA

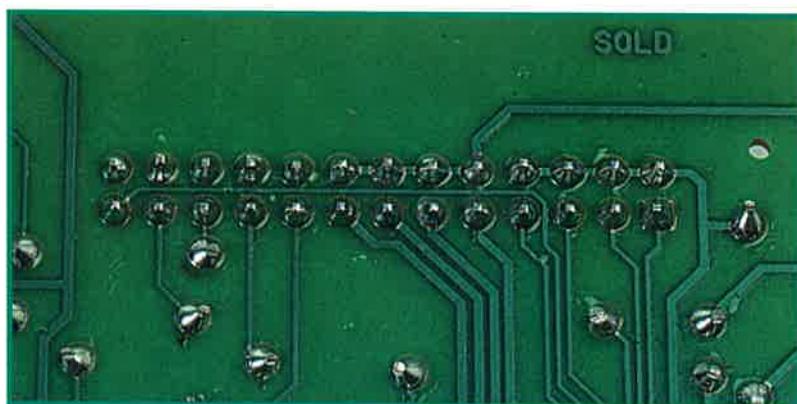
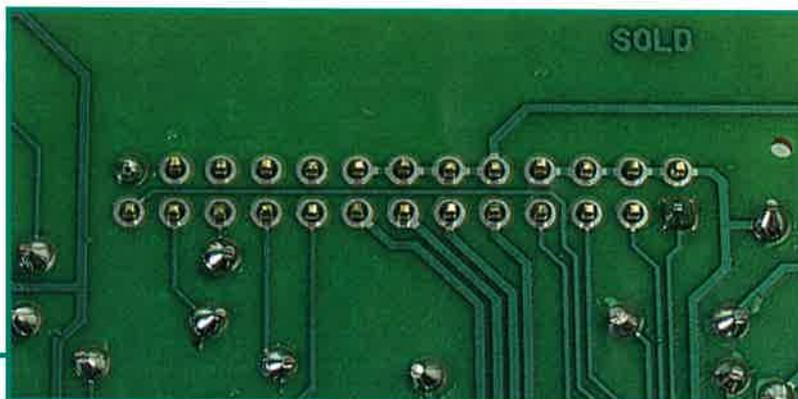
PL 60

Montaggio passo a passo



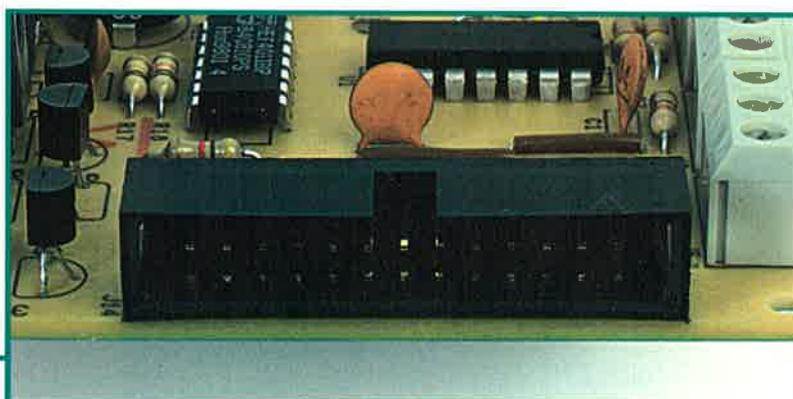
Ora salderemo il connettore CP a 26 vie J14, che servirà per l'inserimento del cavo Picbus, grazie al quale metteremo in comunicazione la scheda di controllo con la scheda di potenza.

Dopo aver montato il connettore, salderemo solo i due piedini agli angoli opposti. Dopo aver fatto queste saldature, verificheremo che il connettore sia totalmente appoggiato sulla superficie della scheda.

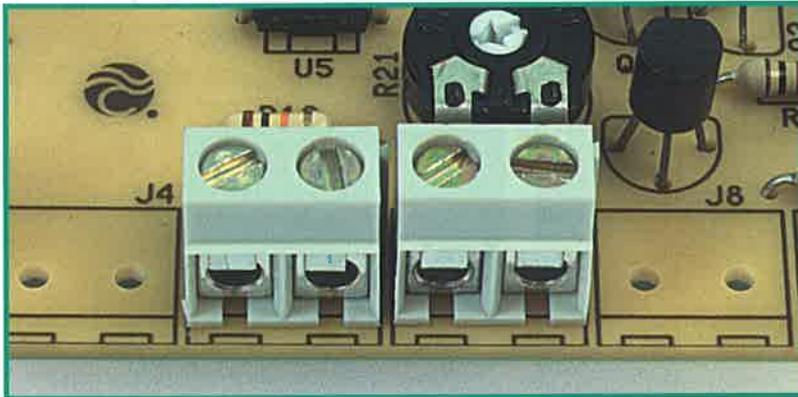


Verificato il corretto inserimento del connettore, termineremo di saldare il resto dei piedini. Dobbiamo mettere poco stagno per evitare che i piedini del connettore entrino in contatto fra di loro.

Questo è l'aspetto che presenta il connettore J14 saldato. Possiamo notare come sia inserito a filo della scheda, pronto per l'inserimento del cavo Picbus.

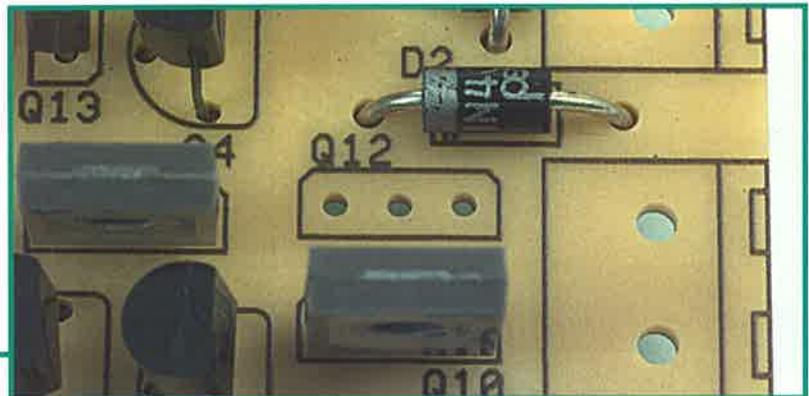


Montaggio passo a passo

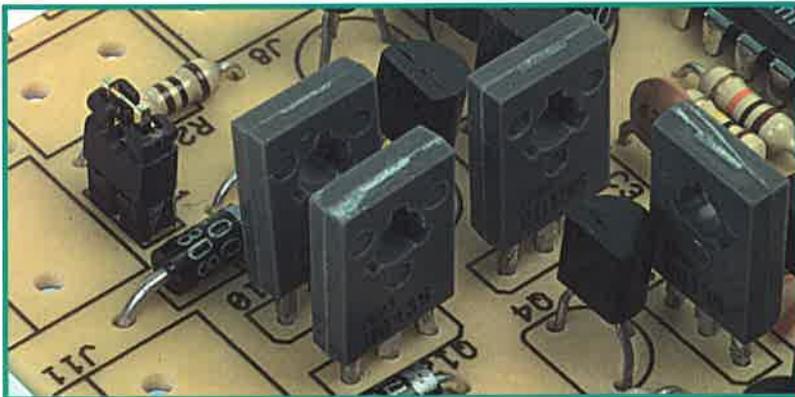


Monteremo e salderemo la morsettiera a due contatti che sarà inserita in J7. La inseriremo totalmente nella scheda e ricopriremo di stagno i suoi piedini sino a formare un cono appoggiato sulla superficie della scheda.

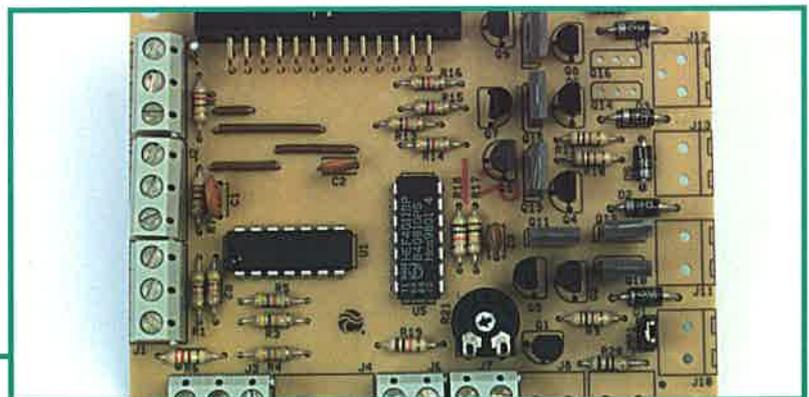
Continueremo saldando il transistor BD135, che verrà inserito in Q12. Ricordate che mediante la serigrafia della scheda possiamo riconoscere qual è il verso di inserzione del transistor.



Questo è l'aspetto del transistor montato e saldato correttamente. È fondamentale posizionare tutti i transistor per il verso corretto, dato che, in caso contrario, potremmo danneggiare la scheda di potenza al momento di metterla in funzione.



Questa è la scheda di potenza con tutti i componenti che abbiamo montato sino ad ora. Manca sempre meno tempo al momento di poter lavorare con essa.



```
c:\progra~1\mplab\esempio1.asm
1 ;*****
2
3     LIST    p-16F04      ;Tipo di dispositivo
4
5     include "P16F04A.INC" ;Definizione dei registri
6
7 ;Registri di utilizzo generale
8
9     CONTATORE    EQU    0x0c
10
```

Realizziamo un programma in cui approfitteremo delle risorse interne del microcontroller: il Watchdog e il modo Sleep. Vogliamo realizzare un modo semplice per evitare i rimbalzi utilizzando queste due risorse, invece di impiegare i temporizzatori, come abbiamo fatto fino ad ora. Per prima cosa selezioniamo il dispositivo e includiamo il file di definizione dei registri.

Il programma consiste in un contatore che utilizza la porta B per portare il risultato su due LED della scheda di ingressi e uscite, e RA0 come ingresso per i segnali da contare (sequenza 0-1-0). In questo caso configuriamo il registro OPTION con il quarto bit a '1' per assegnare il prescaler al Watchdog invece che al Timer0.

```
c:\progra~1\mplab\esempio1.asm
14
15     ORG          INIZIO    0
16     goto
17     ORG          5
18 INIZIO
19     bsf         STATUS, RP0
20     clr         PORTB
21     bsf         PORTA, 0
22     movlw      b'00001111'
23     movwf     OPTION_REG
24     bcf         STATUS, RP0
25
26 ;*****
```

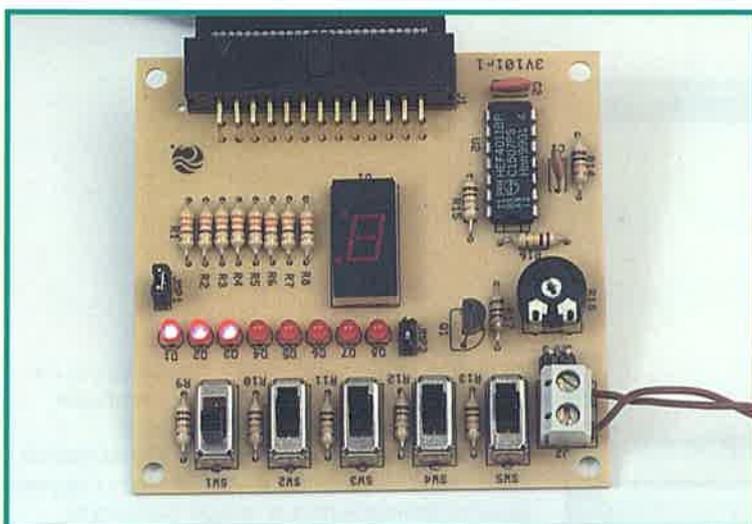
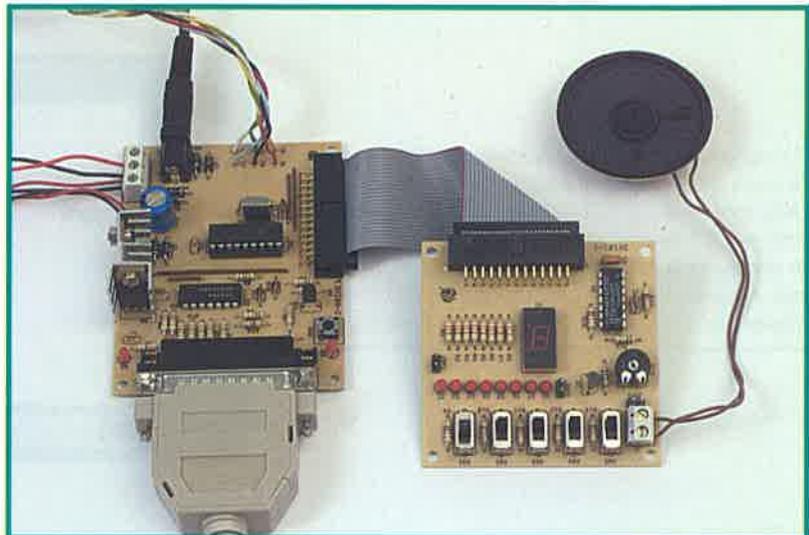
```
c:\progra~1\mplab\esempio1.asm
27
28 CICLO      btfss   PORTA, 0
29            goto   CICLO
30            sleep
31
32 CICLO2     btfsc   PORTA, 0
33            goto   CICLO2
34            sleep
35
36            incf   CONTATORE, F
37            movf   CONTATORE, W
38            movwf  PORTB
39            goto   CICLO
40
41            END            ;Fine del programma
42
43 ;*****
```

Ciclo principale del programma. Ogni volta che c'è un cambio di stato, mandiamo il microcontroller in stato sleep. Quando il Watchdog andrà in overflow, il microcontroller ripartirà, continuando l'esecuzione del programma. A quel punto il segnale di uscita dell'interruttore si sarà già stabilizzato.

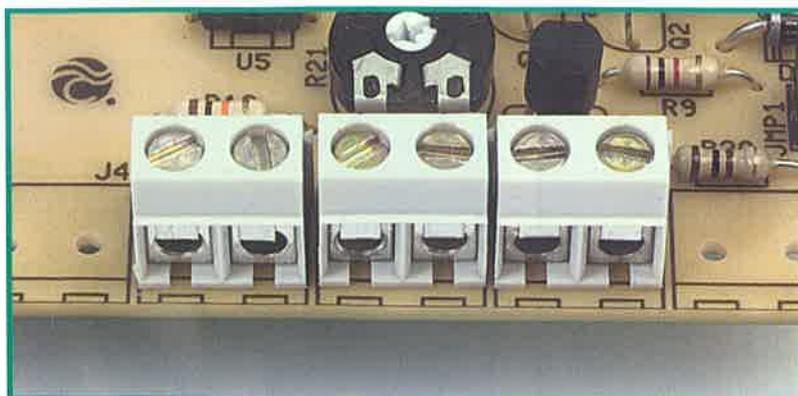


Ricordate di selezionare l'opzione "Watchdog Si" al momento di programmare il microcontroller, dato che questo programma lo utilizza. In questo modo scriveremo la parola di configurazione del PIC, e il programma funzionerà correttamente.

L'attivazione del Watchdog non richiede alcuna modifica per la scrittura del PIC. Realizzeremo questo processo come sempre, collegando la scheda di controllo al PC, con l'interruttore di scrittura nella posizione PROG, e con i jumper della scheda di ingressi e uscite scollegati, nel caso che entrambe le schede siano unite dal cavo Picbus.

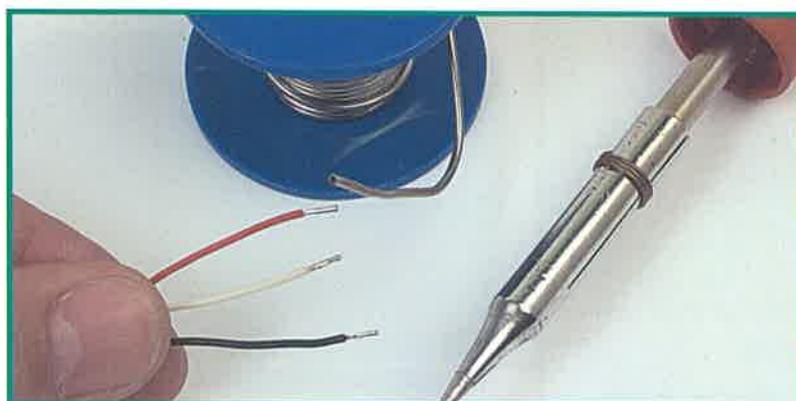
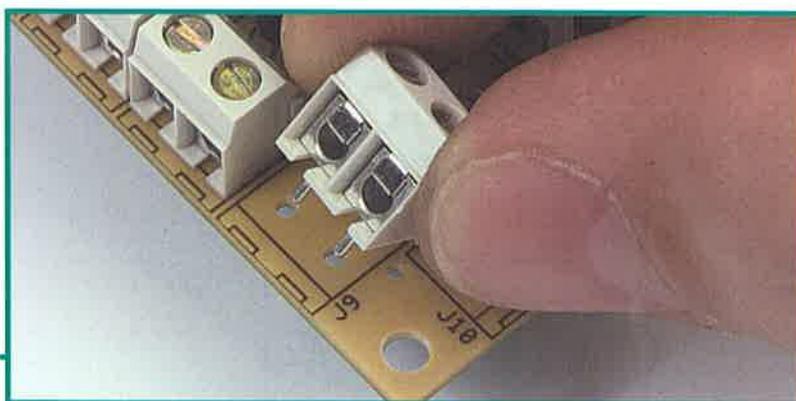


Questa è la scheda di ingressi e uscite con il programma in funzione. Per fare in modo che il dato in uscita si possa vedere sulla barra dei LED, è necessario chiudere il jumper JMP1 della scheda di ingressi e uscite. Ogni volta che introdurremo una combinazione 0-1-0 tramite l'interruttore RA0, si incrementerà il dato di uscita sui LED.



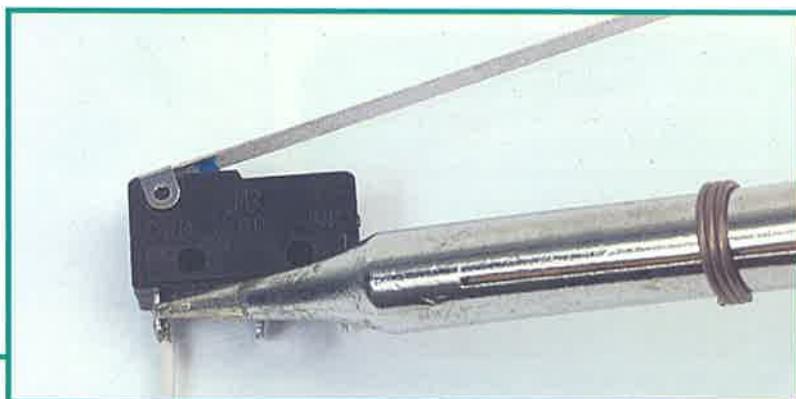
Collegiamo e saldiamo la morsettiere a due contatti che verrà inserita in J8. Dobbiamo assicurarci che la morsettiere sia a filo della scheda e con i collegamenti rivolti verso l'esterno, prima di iniziare a saldare.

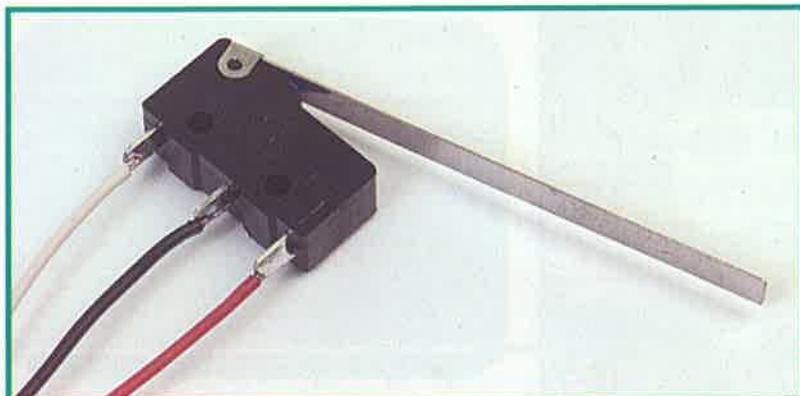
Allo stesso modo inseriremo la morsettiere a due contatti che andrà in J9. Ricordiamo che le saldature delle morsettiere hanno bisogno di una maggiore quantità di stagno rispetto a quello che abitualmente impieghiamo per le altre saldature della scheda.



Ora taglieremo i tre pezzi di cavo a metà; speleremo l'estremità di ognuno dei pezzi, e lo prestagneremo come mostrato nell'immagine.

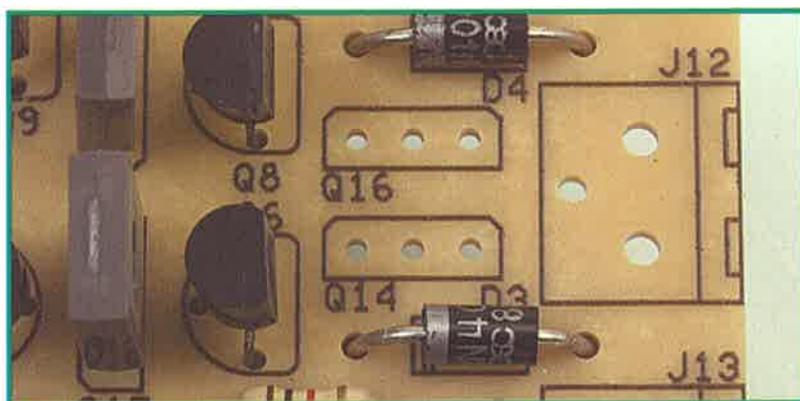
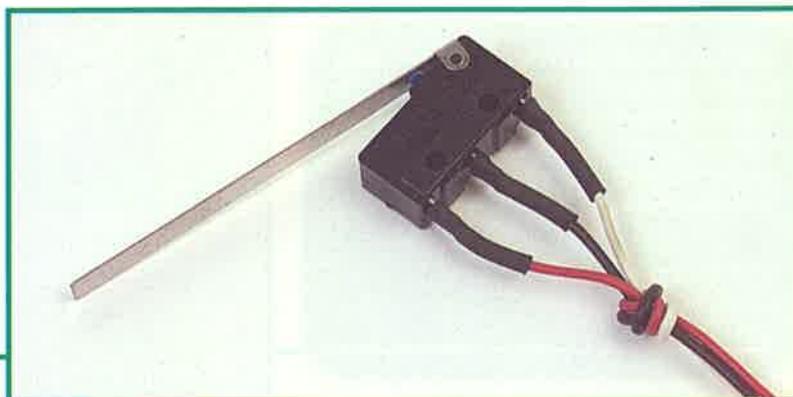
Utilizzeremo un insieme di tre cavi per preparare uno dei finecorsa meccanici. Prestagneremo ognuno dei tre piedini del finecorsa prima di procedere alla saldatura dei cavi.





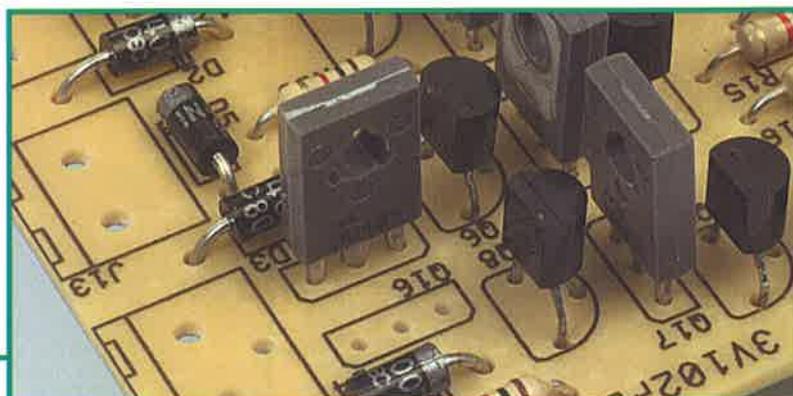
Per saldare ogni cavo al suo piedino dovrebbe essere sufficiente lo stagno che abbiamo già depositato sui piedini e sui cavi. Per questo metteremo in contatto il piedino del sensore con il cavo e riscaldaremo la zona con il saldatore.

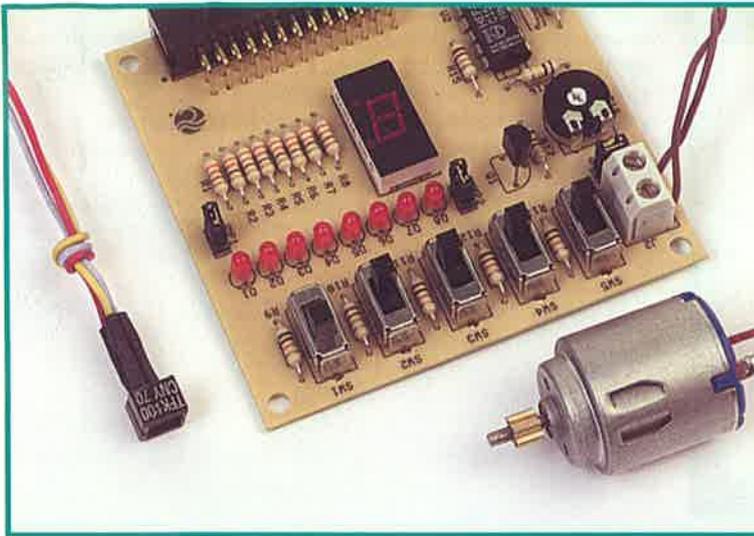
Dopo aver saldato i tre cavi monteremo il termorestringente; inseriremo un pezzo di tubo in ognuno dei tre piedini e poi riscaldaremo la zona con il saldatore, sino a che il termorestringente aderirà alla saldatura.



Monteremo un transistor BD135 in Q14. Per identificare il verso di inserzione sarà sufficiente osservare la serigrafia della scheda.

Questo è il transistor inserito in Q14 e saldato. Ricordate che i transistor non vanno introdotti completamente nella scheda, ma devono risultare sollevati di qualche millimetro dalla superficie della stessa.





Come avrete notato, tutti gli esercizi che abbiamo realizzato con la scheda di ingressi e uscite, sono validi per la gestione dei sensori e dei motori della scheda di potenza. Dato che questa scheda condiziona i segnali per i nostri programmi, non c'è differenza ad inviare un 1 per accendere un LED, o per attivare un motore, ed è la stessa cosa ricevere uno 0 da un interruttore o da un sensore.

Realizziamo un programma di simulazione di movimento del microrobot. Per fare questo utilizzeremo gli interruttori SW1 e SW2 per simulare i segnali di ingresso dei due sensori a infrarossi CNY70 e i LED D1-D4 per simulare le uscite di controllo dei motori di trazione.

```
c:\progra~1\mplab\esempi_2.asm
1 *****
2
3 LIST p=16F84A ;Tipo di dispositivo
4 include "P16F84A.INC" ;Definizione dei registri
5
6 ;Registri di utilizzo generale
7
8 AUX EQU 0x0c
9
10
```

```
c:\progra~1\mplab\esempi_2.asm
10
11 ORG 0
12 goto INIZIO
13 ORG 5
14
15 INIZIO bsf STATUS, RP0
16 clr PORTB
17 movlw b'00000011'
18 bcf STATUS, RP0
19
20
```

Dopo aver scritto l'intestazione del programma, configureremo la porta B come uscita, e come ingressi i corrispondenti degli interruttori SW1 e SW2 della porta A. Non utilizzeremo altri registri del microcontroller (a parte lo STATUS) per questo programma.

```

c:\progra~1\mplab\esempi_2.asm
22
23 CICLO:      movf    PORTA, W
24            andlw   b'00000011'
25            movwf   AUX
26            xorlw   b'00000000'
27            btfsc   STATUS, Z
28            goto    AVANTI
29            movf    AUX, W
30            xorlw   b'00000001'
31            btfsc   STATUS, Z
32            goto    DESTRA
33            movf    AUX, W
34            xorlw   b'00000010'
35            btfsc   STATUS, Z
36            goto    SINISTRA
37            movf    AUX, W
38            xorlw   b'00000011'
39            btfsc   STATUS, Z
40            goto    INDIETRO
41            goto    CICLO

```

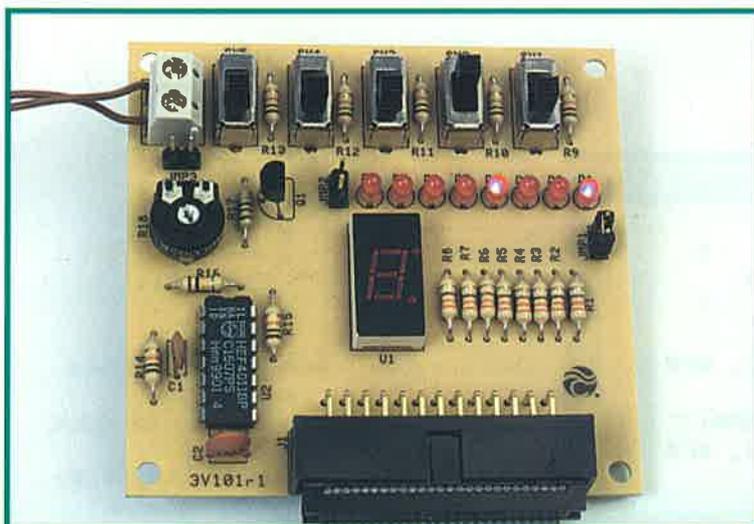
Ogni volta che leggeremo un dato dalla porta A, lo scriveremo su una variabile ausiliaria. In seguito realizzeremo delle maschere per conoscere quale sia il valore dei sensori/interruttori di ingresso. Osserviamo che la maschera AND serve solamente per leggere il dato, e per evidenziare i bit che ci interessano. In seguito realizzeremo delle operazioni XOR e verificheremo il flag Z (zero) del registro STATUS, per eseguire o meno il salto, a seconda delle diverse combinazioni.

Dopo aver decifrato il valore dei sensori, non dovremo far altro che inviare l'ordine corrispondente per la porta B verso i LED: questo equivale ad inviare ordini di controllo verso i motori. I LED D1-D2 simulano le uscite di controllo di un motore, e D3-D4 le uscite di controllo per l'altro motore. I motori possono girare in entrambi i sensi, e la combinazione di questi dà origine ai diversi movimenti. Questo programma, benché semplice ci permetterà di controllare il verso del movimento di Monty. Da questo momento potremo complicare l'algoritmo in funzione della nostra creatività.

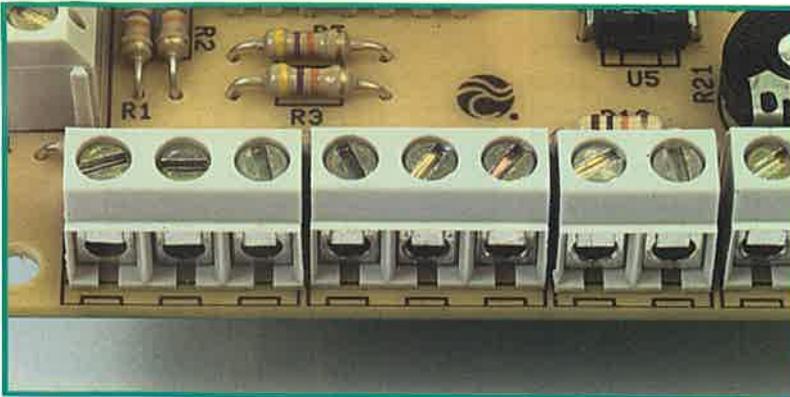
```

c:\progra~1\mplab\esempi_2.asm
44
45 AVANTI      movlw   b'00000011'
46            movwf   PORTB
47            goto    CICLO
48
49 INDIETRO    movlw   b'00001100'
50            movwf   PORTB
51            goto    CICLO
52
53 DESTRA      movlw   b'00000110'
54            movwf   PORTB
55            goto    CICLO
56
57 SINISTRA    movlw   b'00001001'
58            movwf   PORTB
59            goto    CICLO
60
61            END    ;Fine del programma
62
63 ;*****

```

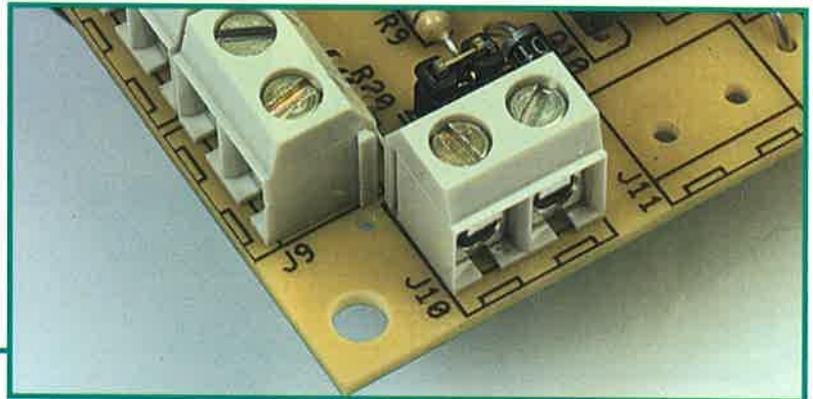


Mostriamo il funzionamento di questo programma sulla scheda di ingressi e uscite. È necessario chiudere il jumper JMP1, per fare in modo che la barra dei LED si possa illuminare, e verificare se gli ordini di uscita che stiamo inviando sono corretti. Vi invitiamo ad eseguire lo stesso esercizio utilizzando però altri interruttori di ingresso, e cambiando gli ordini di uscita della porta B.

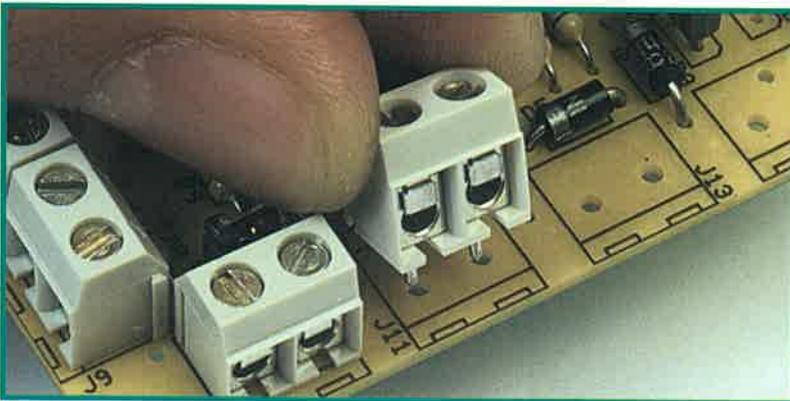


Saldiamo ora l'ultima morsettiera a tre contatti sulla scheda di potenza. Va inserita in J4; dopo averla montata, salderemo i suoi piedini sino ad ottenere dei coni appoggiati sulla superficie della scheda.

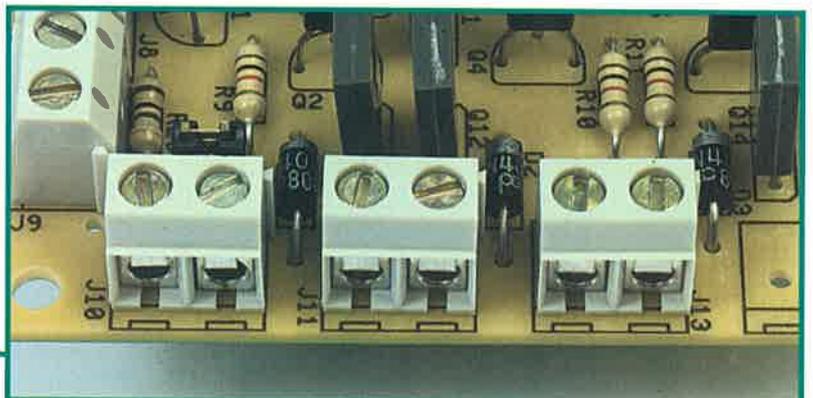
Salderemo ora una morsettiera a due contatti che sarà inserita in J10. Come sempre, i collegamenti della stessa dovranno essere rivolti verso l'esterno della scheda e dovranno essere ben appoggiati sulla superficie.

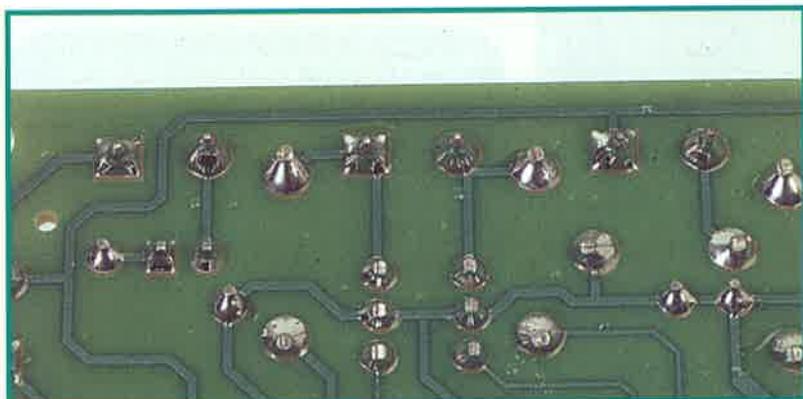


La successiva morsettiera a due contatti va inserita in J11. È importante saldarla correttamente, altrimenti, facendo forza con il cacciavite per avvitarne i morsetti, rischieremo di far saltare via lo stagno.



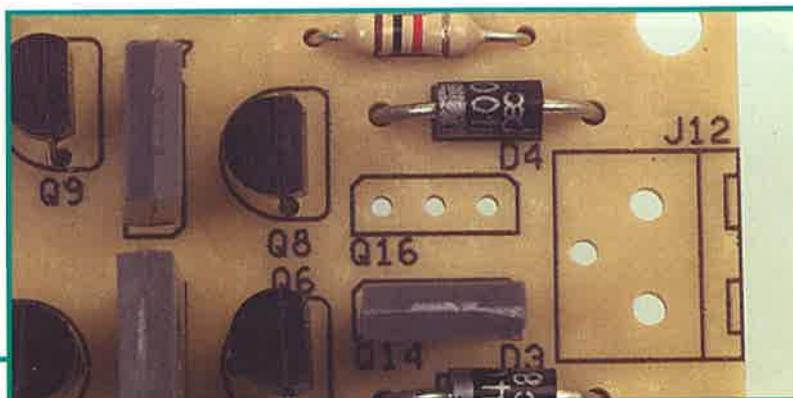
Ora monteremo la morsettiera a due contatti che va inserita in J13. A questo punto ci rimane solamente un'altra morsettiera da montare e avremo terminato il montaggio di tutti gli ingressi e le uscite per la scheda di potenza.



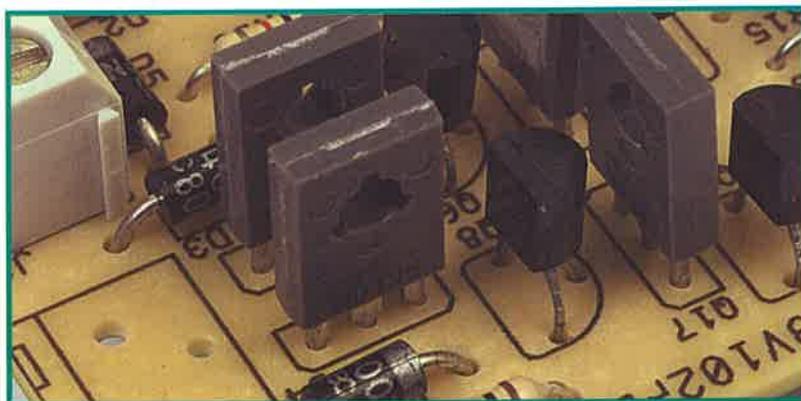


Questo è l'aspetto che devono avere le saldature delle tre morsettiere a due contatti che abbiamo appena saldato. Le saldature, a forma di cono, devono coprire completamente le piazzole e appoggiare sulla scheda.

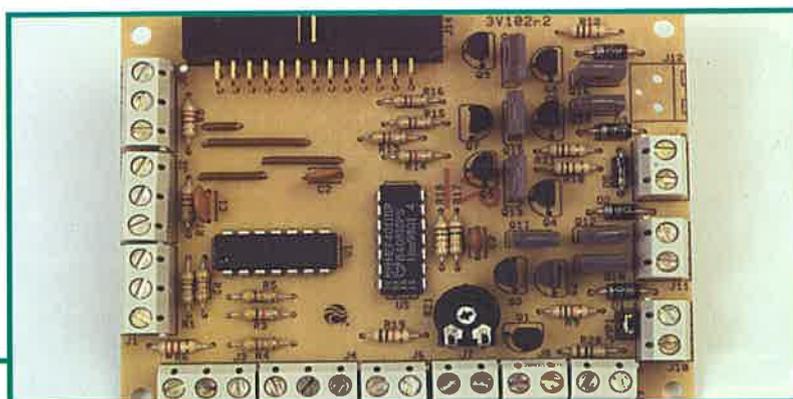
Montiamo ora un transistor BD135. Andrà inserito in Q16. È l'ultimo transistor della scheda di potenza, e con questo terminiamo di montare il circuito di potenza per l'alimentazione dei motori.



Questo è il verso di inserzione del transistor Q16. Dopo averlo saldato raccomandiamo di rivedere il corretto orientamento di tutti i transistor, dato che abbiamo terminato di saldare, e devono essere tutti correttamente montati.



Questo è l'aspetto della scheda di potenza a questo punto del montaggio. Abbiamo già collocato tutti i componenti eccetto una morsettieria a due contatti. Abbiamo anche preparato la maggior parte dei sensori.

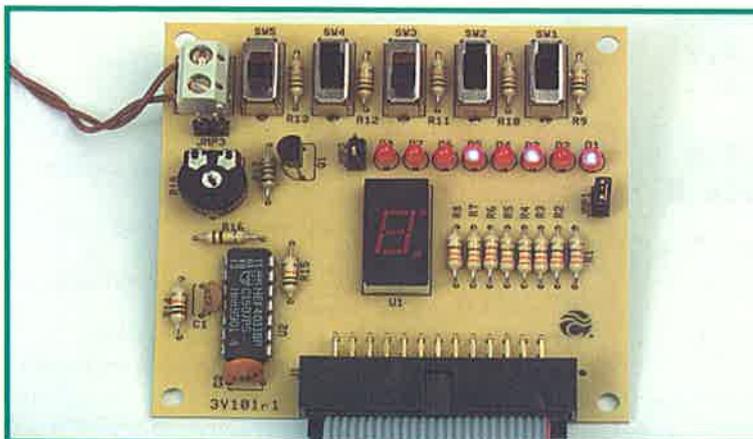


```
c:\progra~1\mplab\asc_33\esempio1.asm
1 ;Equazione: (A+B)-C
2
3
4
5          list    P=16F84A
6          INCLUDE "P16F84A.INC"
7 Dato_A   equ    0x0C
8 Dato_B   equ    0x0D
9 Dato_C   equ    0x0E
10 Risultato equ 0x0F
11
12          org    0x00
13          goto  INIZIO
14
15          org    0x05
16
17 INIZIO   bsf    STATUS, RP0
18          clr    PORTB
19          bcf    STATUS, RP0
20
```

Realizzeremo alcuni esercizi per risolvere dei problemi matematici. Per esempio un'equazione: $(A+B) - C$. Questi valori saranno di 8 bit. Il risultato dell'equazione si mostrerà mediante la porta B e si potrà vedere sulla barra dei LED della scheda di ingressi e uscite. Nell'immagine vediamo l'abituale definizione dei registri.

Il programma configura la porta B come uscita e in seguito carica i dati con i quali si andrà ad operare in tre registri di utilizzo generale: $(6+25)-10$. In seguito eseguiamo le istruzioni aritmetiche necessarie per ottenere il risultato dell'equazione. Infine la risposta si mostra tramite la porta B.

```
c:\progra~1\mplab\asc_33\esempio1.asm
20
21
22          movlw  .6
23          movwf Dato_A
24          movlw  .25
25          movwf Dato_B
26          movlw  .10
27          movwf Dato_C
28          movf  Dato_A,W
29          addwf Dato_B,W
30          movwf Risultato
31          movf  Dato_C,W
32          subwf Risultato,W
33          movf  Risultato,W
34          movwf PORTB
35
36 Ciclo   goto  Ciclo
37
38          END
```



Dopo aver scritto il programma, lo compileremo. Successivamente scriveremo il codice esadecimale nella memoria del microcontrollore utilizzando la scheda di controllo. Una volta scritto il PIC, collegheremo la scheda di controllo con quella di ingressi e uscite attraverso il cavo Picbus. Dobbiamo chiudere il jumper JMP1 per vedere il risultato dell'equazione sulla barra dei LED. Logicamente il dato di risposta si vedrà in numeri binari.

```

c:\progra~1\mplab\fac_33\esempio2.asm
1
2          list    P=16F84A
3          INCLUDE "P16F84A.INC"
4
5 Dato_A_L    equ    0x10
6 Dato_A_H    equ    0x11
7 Dato_B_L    equ    0x12
8 Dato_B_H    equ    0x13
9 Risultato_L equ    0x14
10 Risultato_H equ    0x15
11
12          org    0x00
13          goto   INIZIO
14
15          org    0x05
16

```

Questo esercizio è stato semplice perché abbiamo usato operandi da 8 bit, però questi ci permettono di lavorare solo con numeri fino al valore di 255 come massimo. Le operazioni con numeri da 16 bit sono più complesse, dato che non esiste un'istruzione diretta del compilatore che li possa realizzare. In seguito esporremo come sommare e sottrarre numeri da 16 bit. L'immagine mostra la classica definizione dei registri utilizzati.

L'insieme di istruzioni dell'immagine serve per sommare due registri da 16 bit. La parte segnata con *_L* (low) in ogni operando rappresenta la parte bassa di ogni registro da 16 bit. La parte alta è simbolizzata aggiungendo *_H* (high) ad ogni registro. Come si può vedere, anche il risultato della somma si memorizza su un registro da 16 bit.

```

c:\progra~1\mplab\fac_33\esempio2.asm
16
17 INIZIO    movf    Dato_A_L,W
18          addwf   Dato_B_L,W
19          movwf   Risultato_L
20          movf    Dato_A_H,W
21          btfsc   STATUS,C
22          addlw   1
23          addwf   Dato_B_H,W
24          movwf   Risultato_H
25
26 Stop     nop
27          nop
28
29          end

```

```

c:\progra~1\mplab\fac_33\esempio3.asm
INIZIO    movf    Dato_B_L,W
          subwf   Dato_A_L,W
          movwf   Risultato_L
          movf    Dato_B_H,W
          btfss   STATUS,C
          addlw   1
          subwf   Dato_A_H,W
          movwf   Risultato_H

Stop     nop
          nop
          end

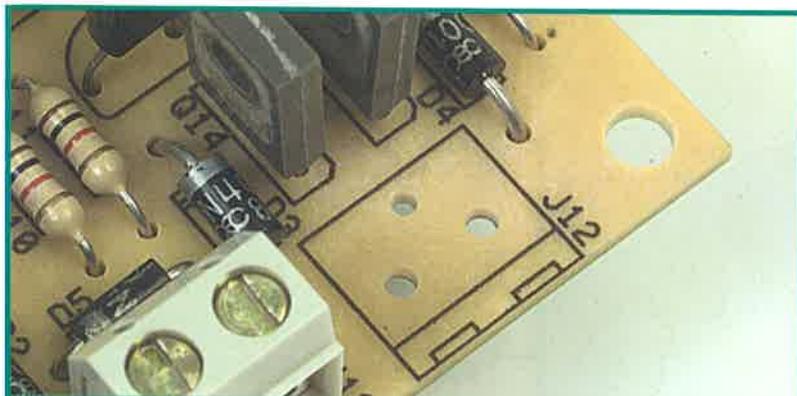
```

L'insieme di istruzioni in figura, serve per sottrarre due numeri da 16 bit. Conviene scrivere e provare queste routines, che memorizzeremo nei nostri file, dato che potrebbero essere utili in qualsiasi momento, durante la realizzazione di algoritmi di controllo con il robot, in cui sia necessario maneggiare registri da 16 bit. Sarà sufficiente aprire le nostre routines matematiche, già provate, ed includerle nel nostro programma.

MODULO DI POTENZA

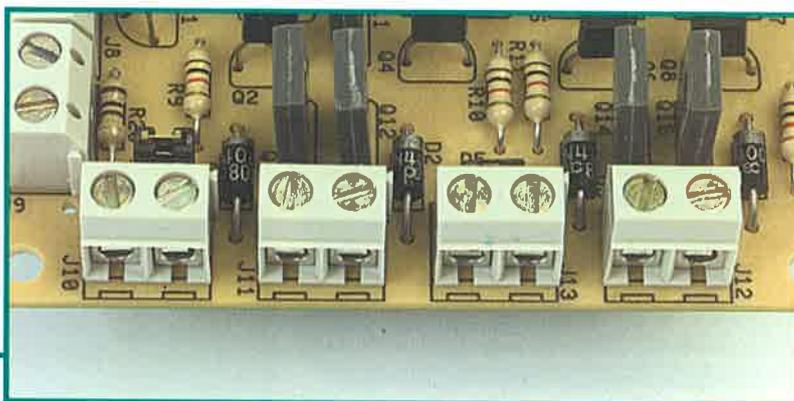
PL 66

Montaggio passo a passo

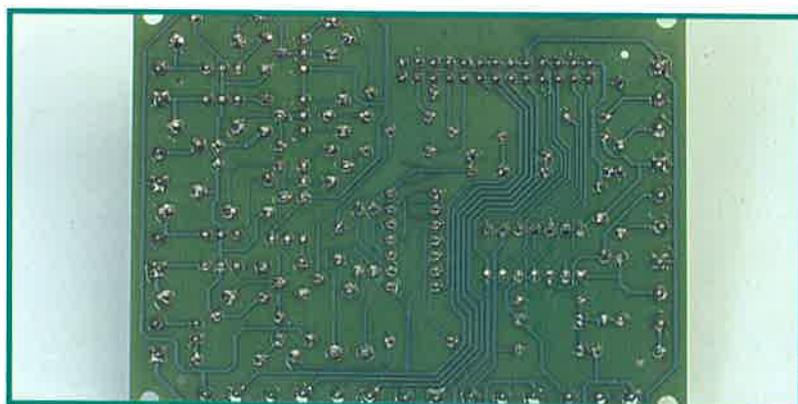


Dobbiamo saldare l'ultimo componente di cui è composta la scheda di potenza. Si tratta di una morsetteria a due contatti che sarà collocata in J12.

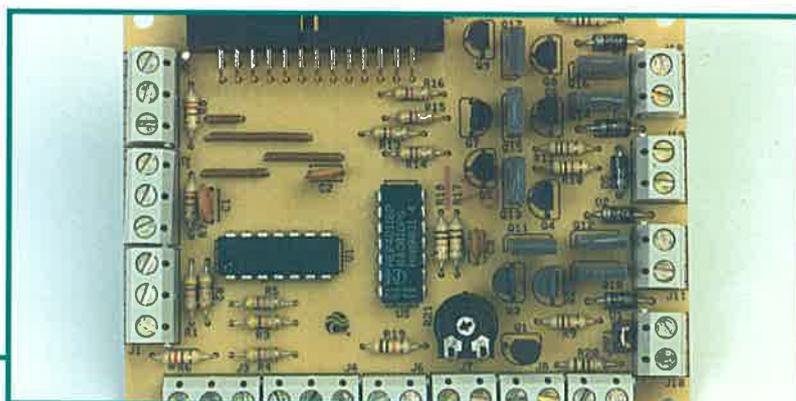
Introdurremo la morsetteria nel verso mostrato dall'immagine fino a che sia totalmente a filo della scheda. Dopo volteremo la scheda e procederemo alla saldatura.



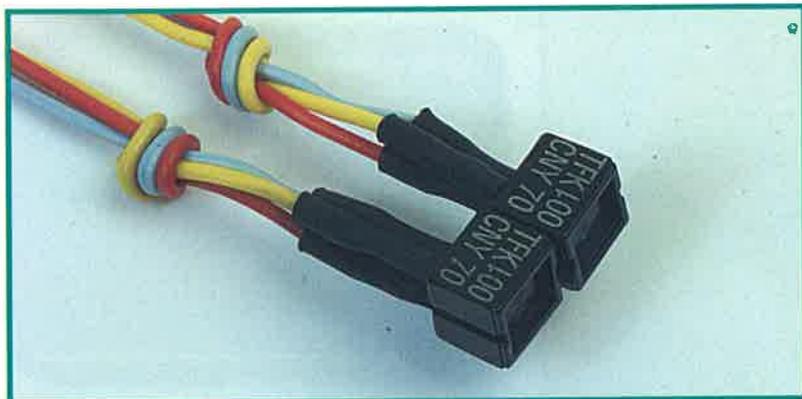
A questo punto abbiamo montato e saldato tutti i componenti della scheda di potenza. Questo è l'aspetto che devono presentare le saldature: brillanti e a forma di cono. È il momento di controllarla per trovare eventuali difetti.



Aspetto della scheda di potenza con tutti i suoi componenti saldati e con i ponticelli necessari realizzati. Verificate che i componenti siano nel posto adeguato e con l'orientamento corretto.

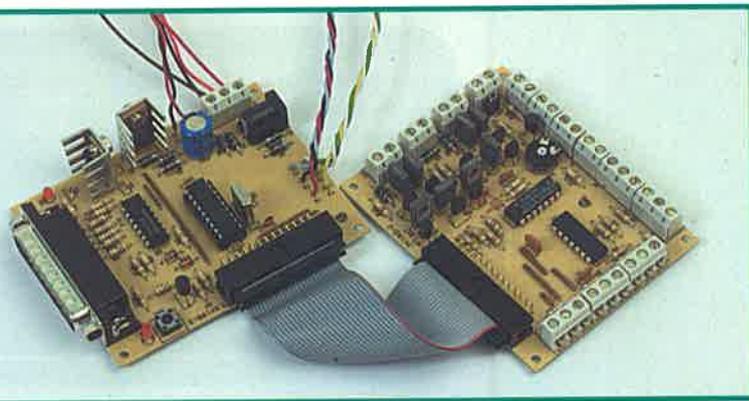
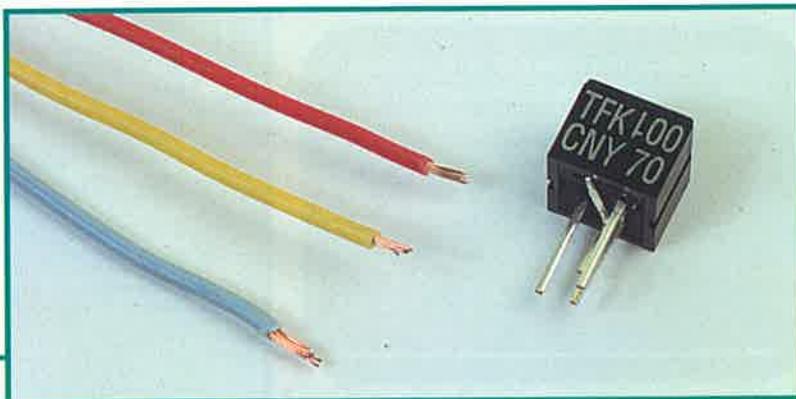


Montaggio passo a passo



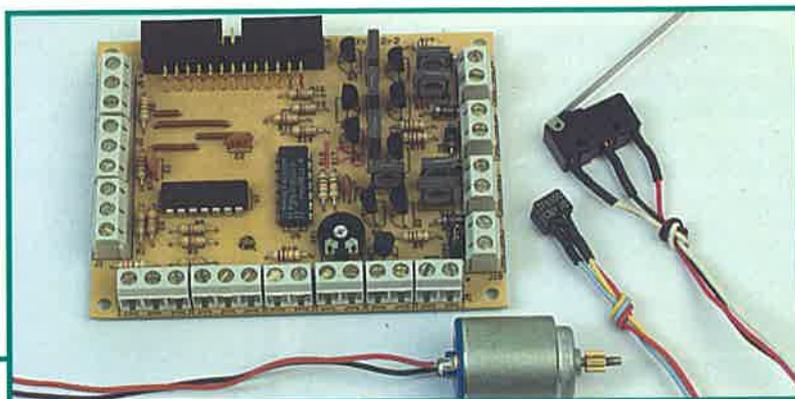
Prepariamo ora un altro sensore ad infrarossi CNY70. Con sensori ottici uniti, come mostrato dall'immagine, renderemo Monty capace di seguire la traiettoria di una linea nera dipinta sul pavimento.

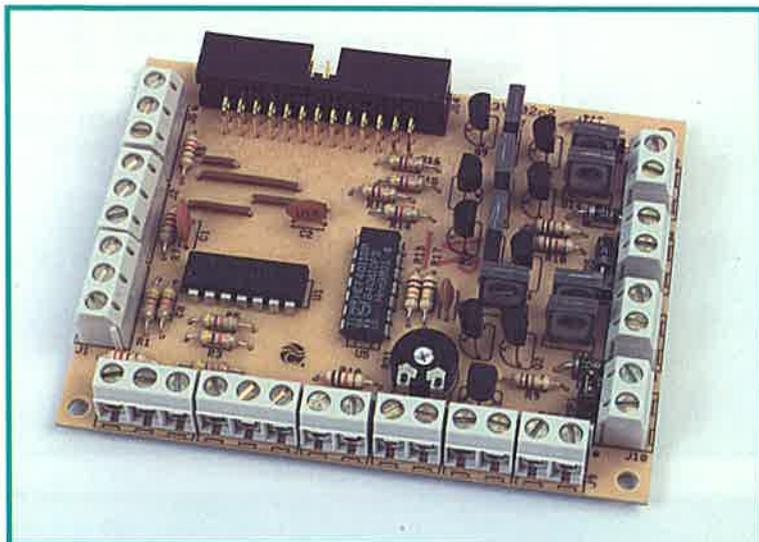
Eseguiamo la preparazione del sensore nel modo che abbiamo già visto. Salderemo tre cavi (dobbiamo unire i due piedini corrispondenti del sensore), e dopo monteremo il termorestringente, per evitare possibili cortocircuiti.



La scheda di potenza comunicherà con la scheda di controllo tramite il cavo Picbus, nello stesso modo in cui realizzavamo la comunicazione con la scheda di ingressi e uscite quando provavamo gli esercizi.

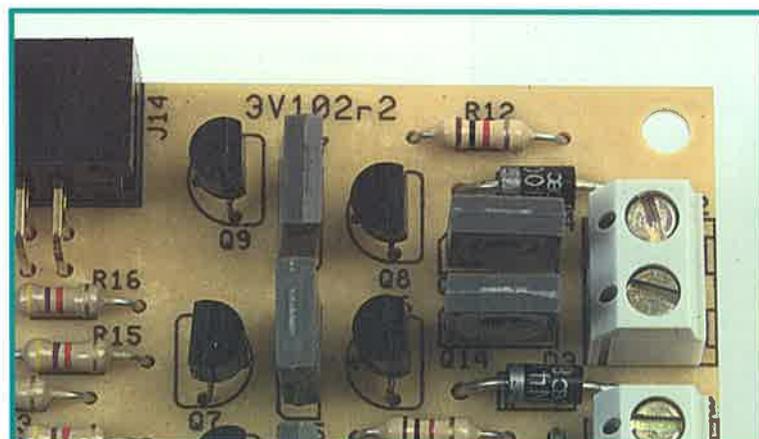
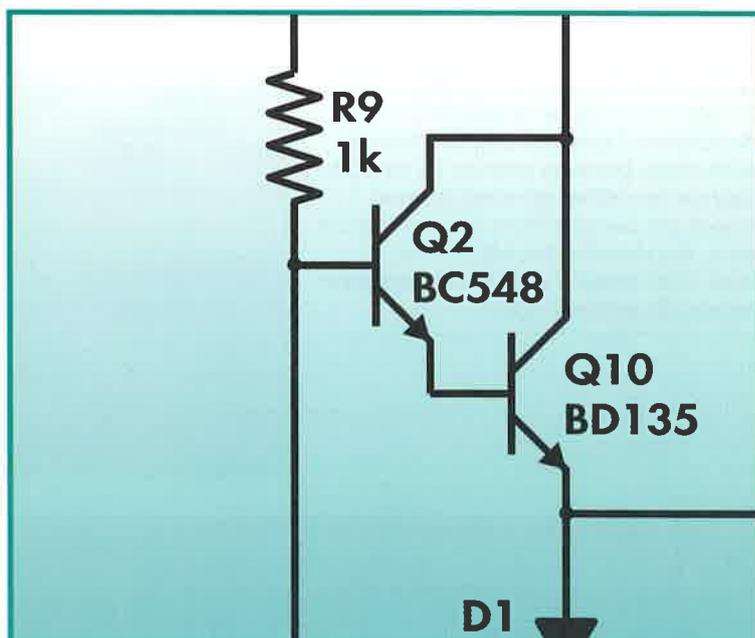
A differenza della scheda di ingressi e uscite, la scheda di potenza ci servirà per gestire altri tipi di periferiche di ingresso, quali i sensori di presenza ottici e/o meccanici al posto degli interruttori, e come periferiche di uscita potremo pilotare, fra le altre cose, i motori di Monty.



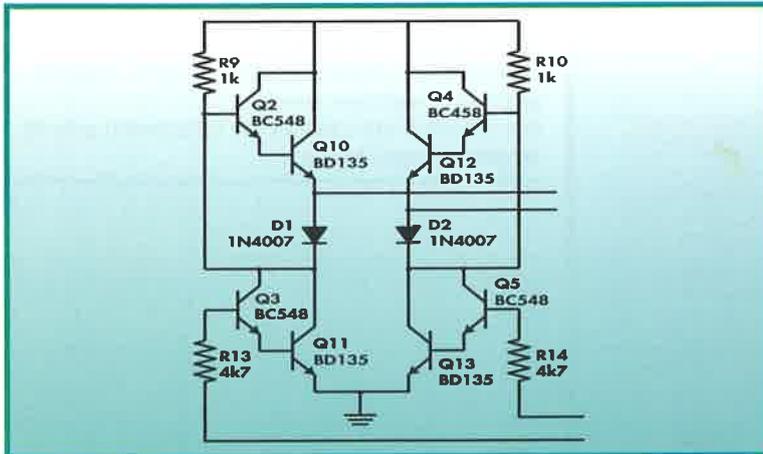


Terminato il montaggio della scheda di potenza, passiamo alla spiegazione delle parti di cui è composto il suo circuito, in modo da imparare a gestirla per realizzare esercizi con i sensori e con i motori che utilizzeremo quando programmeremo Monty.

I segnali che genera il microcontroller PIC sulle sue uscite sono di livello TTL, cioè di 5 V. La capacità di corrente di ogni piedino è di 20 mA. Questo non è sufficiente per alimentare un motore, pertanto non potremo collegarlo direttamente. Sulla scheda di potenza abbiamo implementato un circuito denominato ad H, che si basa sui transistor Darlington che corrispondono allo schema riportato in figura.

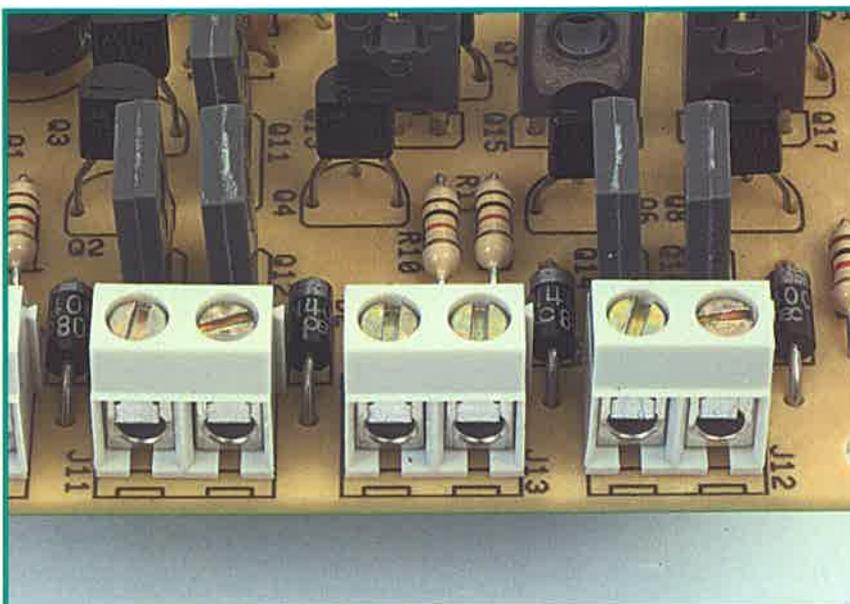
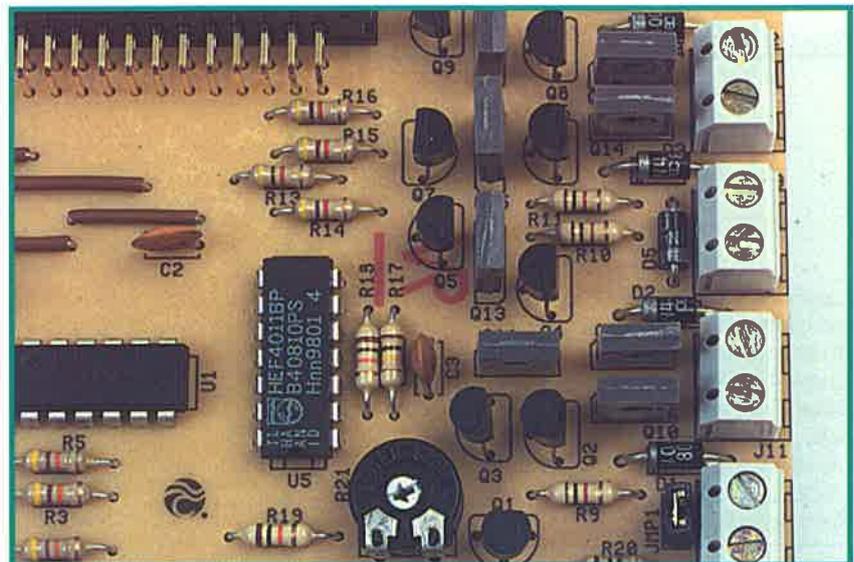


Un transistor Darlington è formato dall'associazione di due transistor, in modo che il guadagno di entrambi si sommi e si ottenga una maggiore capacità di carico rispetto a un solo transistor. Come mostrato nell'immagine, nella nostra scheda la configurazione Darlington è stata implementata con una coppia di transistor formata da un transistor modello BC548 e un BD135.



Questo è lo schema corrispondente al ponte ad H che si utilizza per il controllo di un motore. Due segnali del microcontrolloer pilotano le basi di Q3 e di Q5. Se si attiva la base di Q3 ma non quella di Q5, si attivano i Darlington formati da Q3-Q11 e da Q4-Q12: il motore gira in un verso. Attivando la base di Q5 ma non quella di Q3, si attivano i Darlington formati da Q5-Q13 e da Q2-Q10: il motore gira nel verso contrario. Se si attivano o disattivano simultaneamente le basi di Q3 e di Q5, il motore si fermerà. Il motore sarà collegato ai due emettitori di Q10 e Q12. Il ponte ad H si alimenta con la tensione che si applica al motore, e che nel nostro caso arriverà da un alimentatore diverso da quello utilizzato per alimentare l'elettronica.

Sulla scheda di potenza abbiamo implementato due ponti ad H, ognuno per controllare uno dei due motori che compongono il sistema mobile di Monty. In questo modo possiamo ottenere un controllo totale sul movimento del robot, facendo in modo che avanzi, si fermi, giri a sinistra o a destra, descrivendo curve più o meno chiuse. Nell'immagine vediamo i 16 transistor che formano entrambi i ponti.

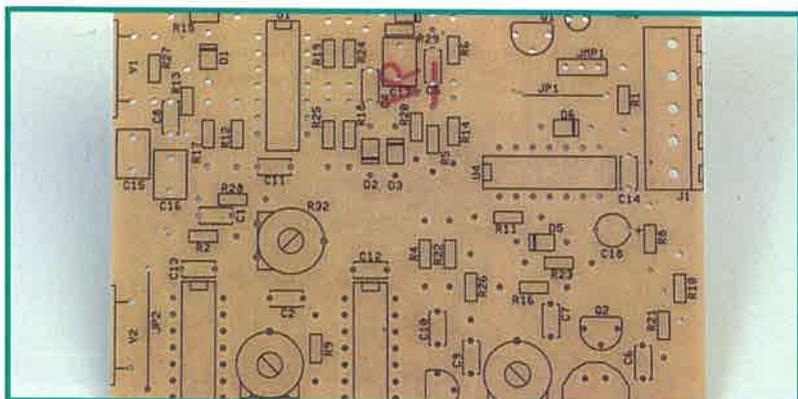


La morsettiera J13, sarà quella che utilizzeremo come ingresso per la tensione di alimentazione dei motori. J11 e J12 serviranno per collegare questi motori al sistema mobile del robot. Come si può vedere, la morsettiera di ingresso dell'alimentazione per i motori ha due morsetti, dato che è necessario collegare anche la massa dell'alimentazione, perché questa deve essere comune con la massa dell'elettronica. Tutte le masse di un sistema elettronico debbono essere comuni, anche se arrivano da alimentatori diversi.

MODULO DEI SENSORI

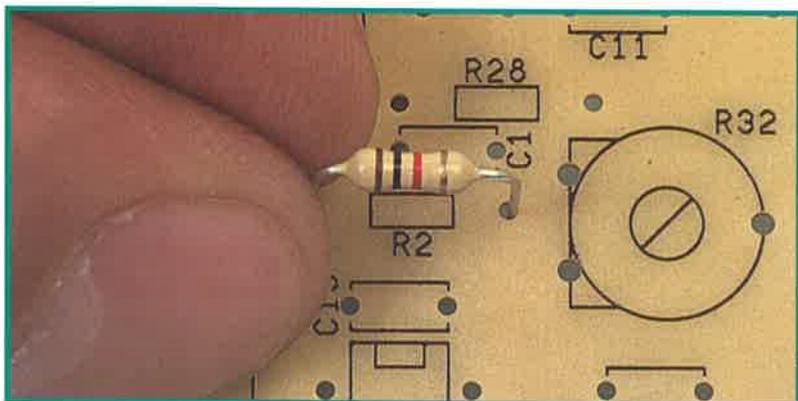
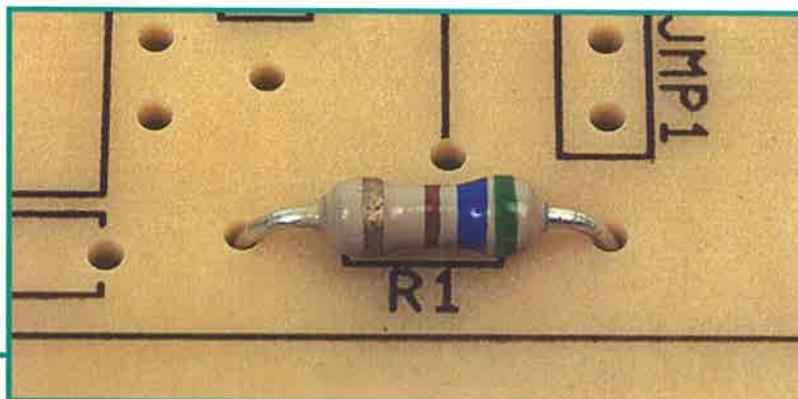
PL 68

Montaggio passo a passo



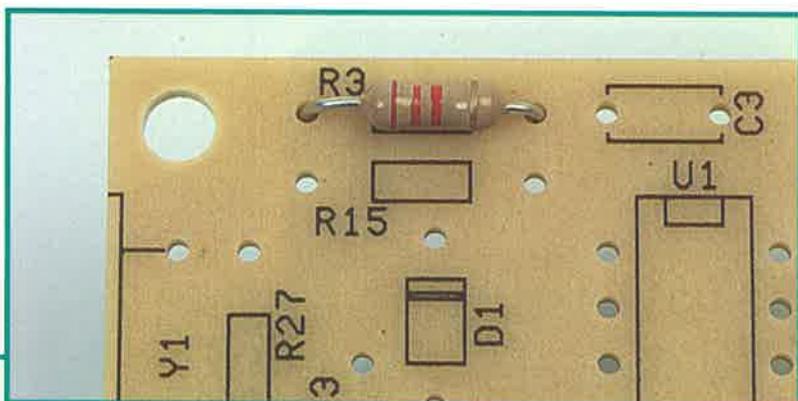
Questa è la nuova scheda che monteremo. La chiameremo scheda dei sensori e servirà per gestire sensori più sofisticati, come sensori ad ultrasuoni per la rilevazione del movimento, o presenza e rilevatori di suoni.

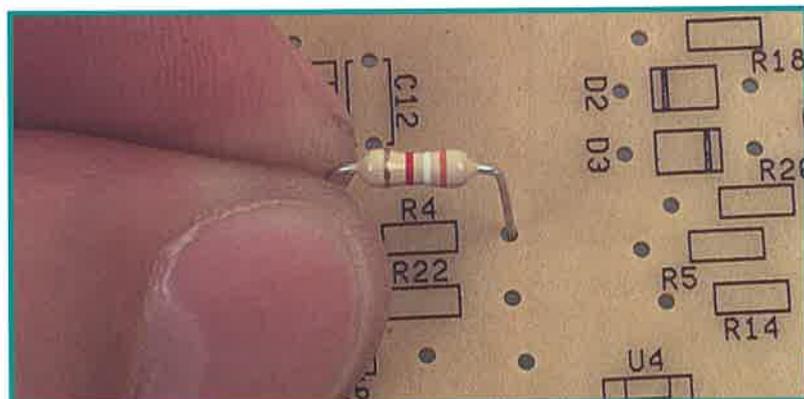
Il primo componente che monteremo sarà la resistenza R1. Il suo valore è di 560 Ω e il suo codice a colori è verde-blu-marrone.



Ora salderemo una resistenza da 1 K Ω . Il suo codice a colori è marrone-nero-rosso. La inseriremo e la salderemo nella posizione R2.

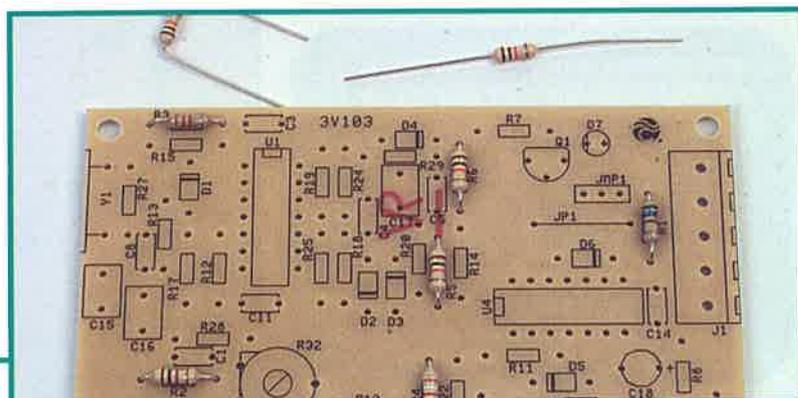
Passiamo ora a saldare una resistenza da 2,2 K Ω . Il suo codice a colori è rosso-rosso-rosso. Va inserita in R3. La monteremo nello stesso verso con cui abbiamo inserito R2, per questioni di estetica.



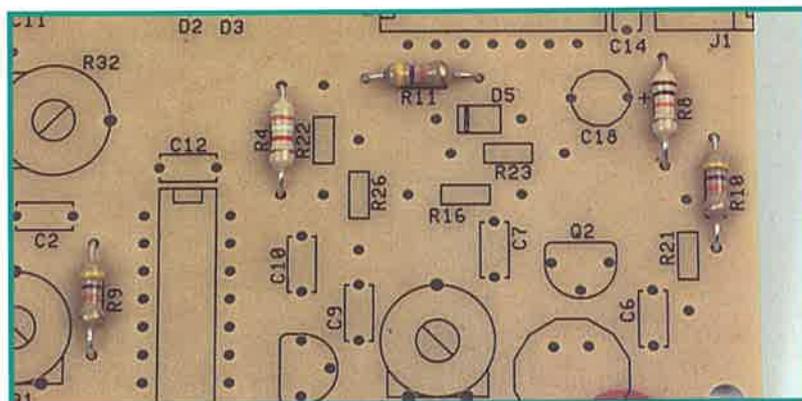


La successiva resistenza da saldare, rispettando il verso di inserzione delle precedenti, è R4. Il suo valore è di 3,9 K Ω e il suo codice a colori è arancio-bianco-rosso.

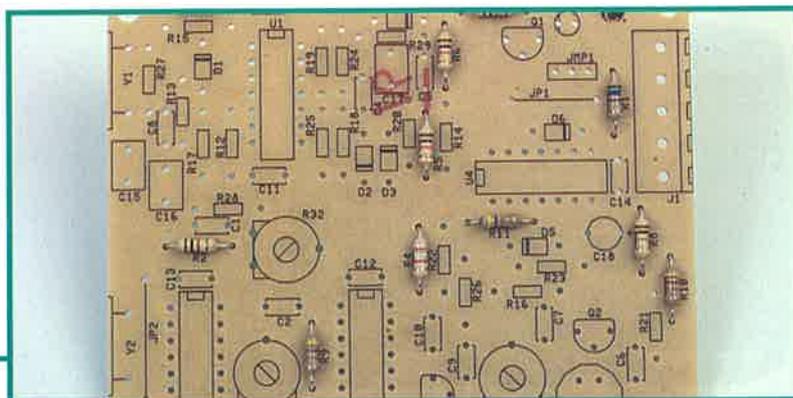
Saldiamo ora quattro resistenze da 10 K Ω . Come abbiamo già visto il codice a colori di queste resistenze è marrone-nero-arancio. Vanno inserite in R5, R6, R7 e R8.



Infine salderemo tre resistenze da 47 K Ω . Ricordiamoci di seguire lo stesso verso di inserzione delle resistenze precedenti. Il codice a colori di queste resistenze è giallo-viola-arancio, e vanno in R9, R10 e R11.



Ecco come deve rimanere la scheda dei sensori, dopo aver saldato questa serie di resistenze. Tutte devono appoggiare sulla superficie della scheda. Le saldature dovranno essere brillanti e lo stagno sufficiente.



```

c:\progra~1\mplab\asc_35\esempio1.asm
1
2 List p=16F84a
3 include "P16F84a.INC"
4
5 org 0x00
6 goto INIZIO
7
8 org 0x05
9
10 INIZIO clrf PORTB
11 bsf STATUS,RP0
12 clrf TRISB
13 movlw b'00011111'
14 movwf TRISA
15 bcf STATUS,RP0
    
```

Faremo un esempio di automatismo sequenziale, in cui diversi eventi devono susseguirsi uno dietro l'altro. L'enunciato è il seguente: la sequenza inizia premendo "I" (RA0), che attiva il cilindro "V" (RB0); quest'ultimo avanzando spinge il pezzo da tornire e aziona il sensore "b" (RA2) attivando il relè del motore "M" (RB1). Inizia la tornitura. Durante l'avanzamento si arriva sul sensore "c" (RA3) che disattiva il cilindro "V" (RB0) iniziando la retrocessione del medesimo. Passando nuovamente su "b" (RA2) si scollega "M" (RB1). Quando si arriva al fine corsa "a" (RA1) si attiva un segnale acustico "A" (RB2) per fare in modo che l'operaio ritiri il pezzo, ne monti un altro nuovo e inizi un nuovo ciclo premendo "I" (RA0). L'immagine mostra la sequenza di inizio, da dove si programmano le porte.

I sensori sono simulati dagli interruttori della scheda di ingressi e uscite. Il cilindro, il motore e il segnale acustico si simulano mediante i LED. È molto importante fare bene i programmi sequenziali, dato che molti dei compiti che assegneremo a Monty saranno di questo tipo. Il programma dell'immagine controlla la sequenza in cui il cilindro si attiva e avanza.

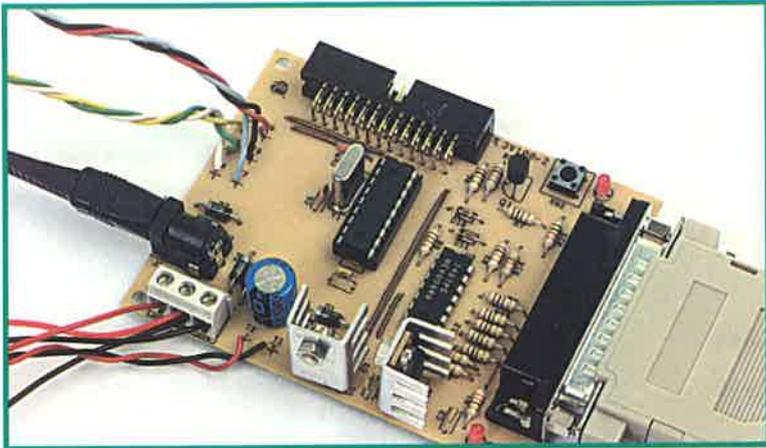
```

c:\progra~1\mplab\asc_35\esempio1.asm
16
17 CICLO: clrwtd
18 btfss PORTA,0
19 goto CICLO
20 bcf PORTB,2
21 bsf PORTB,0
22
23 Attendi_b_1 clrwtd
24 btfss PORTA,2
25 goto Attendi_b_1
26 bsf PORTB,1
27 Attendi_c clrwtd
28 btfss PORTA,3
29 goto Attendi_c
30 bcf PORTB,0
    
```

```

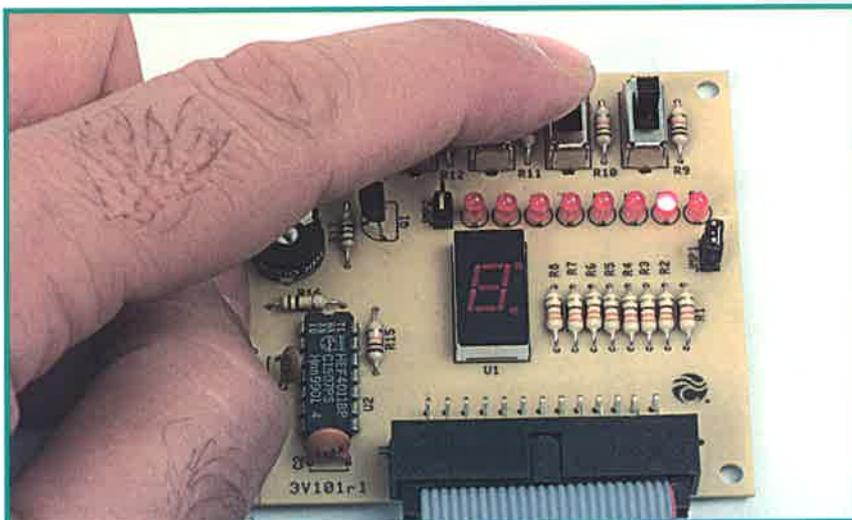
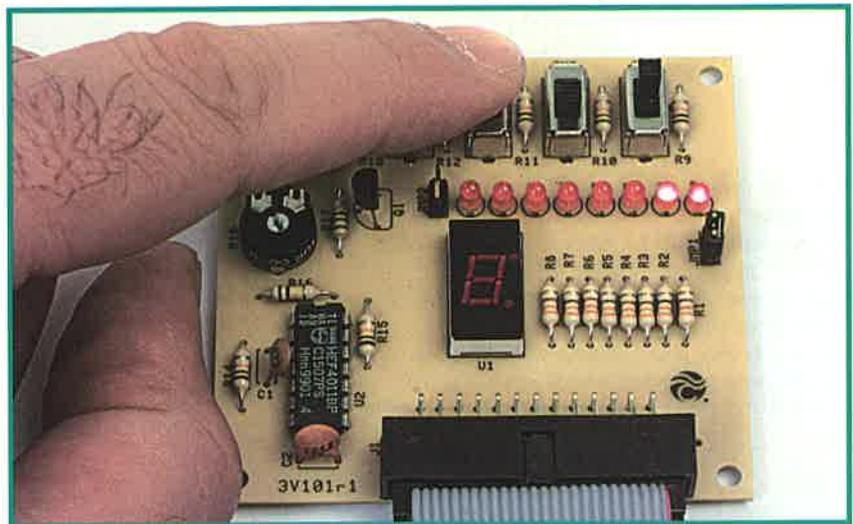
c:\progra~1\mplab\asc_35\esempio1.asm
31
32 Attendi_b_2 clrwtd
33 btfss PORTA,2
34 goto Attendi_b_2
35 bcf PORTB,1
36
37 Attendi_a clrwtd
38 btfss PORTA,1
39 goto Attendi_a
40 bsf PORTB,2
41 goto CICLO
42
43 end
    
```

Questo è l'ultimo ciclo del programma, in cui il cilindro si disattiva e inizia la sequenza di retrocessione. Come si può vedere abbiamo seguito ordinatamente i compiti da realizzare, controllando lo stato delle uscite e dividendo il programma in diversi cicli, in ognuno dei quali si realizza una parte delle sequenze, e si attende un evento per passare ad eseguire quella successiva.



Compileremo il programma e lo scriveremo sul microcontroller utilizzando la scheda di controllo, come già sappiamo fare. Ricordiamo che l'interruttore di scrittura deve stare nella posizione PROG e non RUN. Scollegate la scheda di controllo dal Picbus prima di programmare il PIC. Dopo aver eseguito la programmazione togliete il cavo di connessione al PC dalla scheda di controllo, collegate il cavo Picbus con la scheda di ingressi e uscite e infine chiudete il jumper JMP1 per fare in modo che i LED possano essere attivati.

La prima cosa che dobbiamo fare è mettere l'interruttore RA0 a '1' per iniziare la sequenza e attivare il cilindro V, come recita l'enunciato. In seguito attiveremo l'interruttore RA2 per simulare che il cilindro ha raggiunto il sensore b lungo il suo percorso. Dopo si attiverà il motore, segnale RB1.

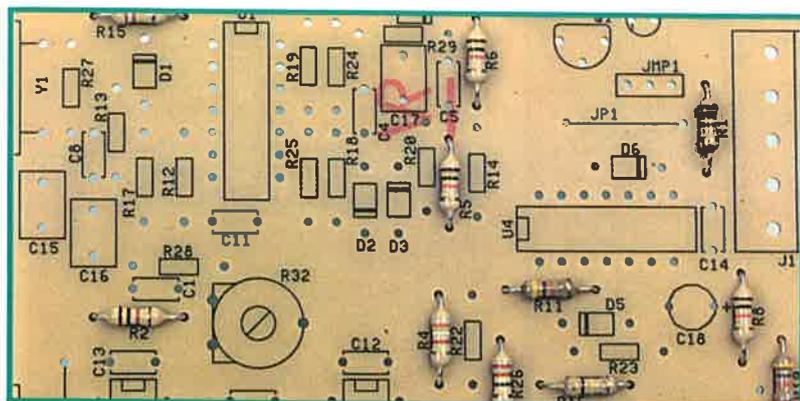


Continueremo in questo modo per verificare la sequenza del programma che abbiamo implementato, e per vedere se il funzionamento sul circuito corrisponde in modo fedele a quanto programmato. Invitiamo i lettori a inventare prove e a realizzare esercizi sequenziali, dato che sono semplici, ed è molto importante padroneggiarli, perché saranno la base per costruire sequenze più complesse per il funzionamento di Monty.

MODULO DEI SENSORI

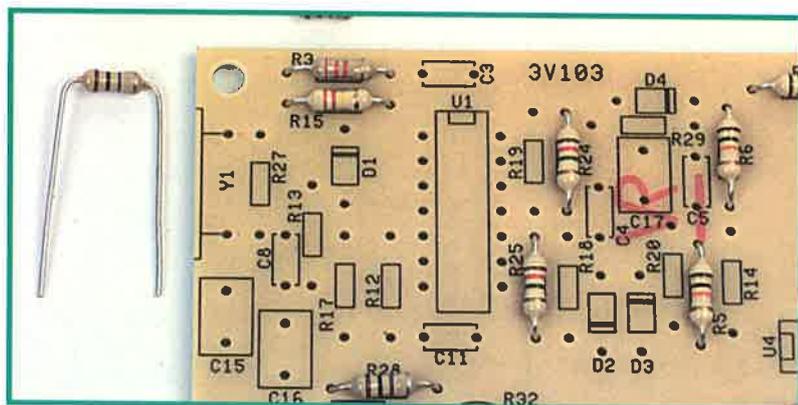
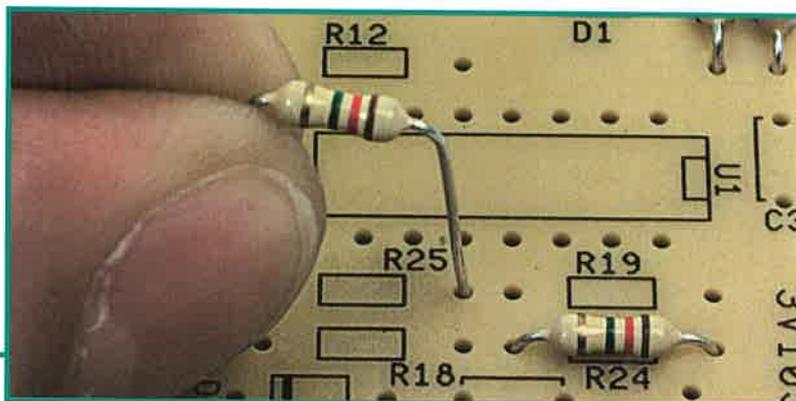
PL 70

Montaggio passo a passo



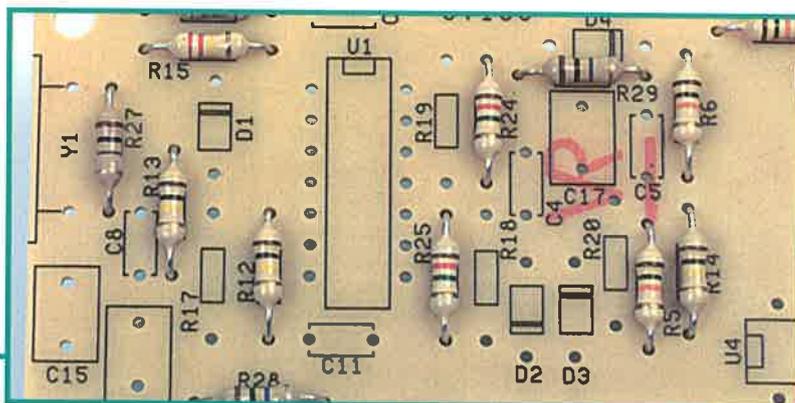
Salderemo tre resistenze. In R16 inseriremo quella da 470 K Ω (giallo-viola-giallo), in R26 da 2M2 (rosso-rosso-verde) e in R15 da 220 K Ω (rosso-rosso-giallo).

Salderemo ora due resistenze da 1M2. Il loro codice a colori è marrone-rosso-verde e vanno inserite in R24 e R25.

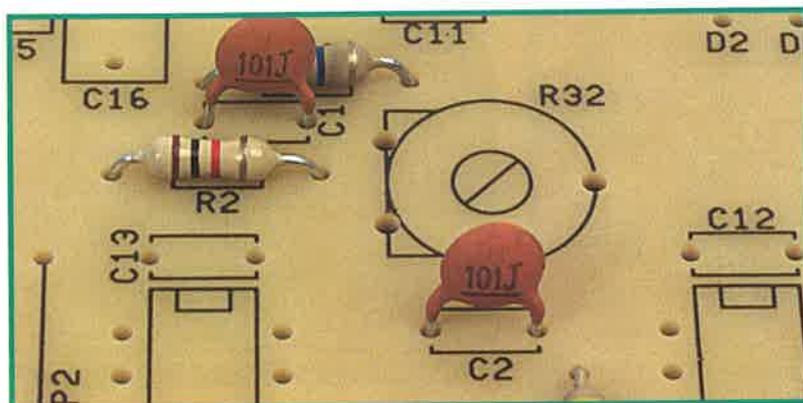


Ora monteremo altre tre resistenze, il cui valore è 10 M Ω . Vanno saldate in R27, R28 e R29; il loro codice a colori è marrone-nero-blu.

Continueremo saldando tre resistenze da 100 K Ω . Il loro codice a colori è marrone-nero-giallo, e vanno saldate in R12, R13 e R14. Dobbiamo fare attenzione a rispettare il verso di inserzione delle resistenze per questioni di estetica: la banda dorata va sempre dalla stessa parte.

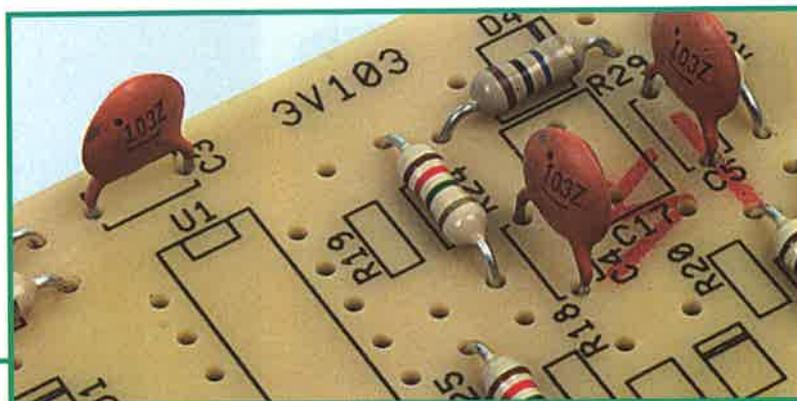


Montaggio passo a passo

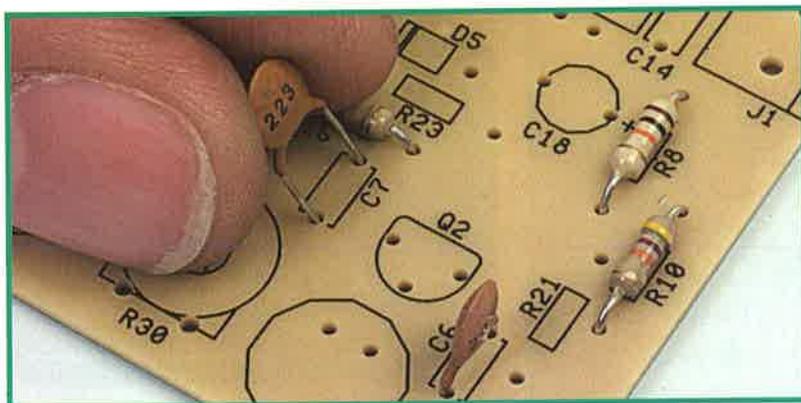


Ora salderemo i condensatori ceramici. Questo tipo di condensatori non ha polarità, quindi non importa il verso con cui li inseriremo. Salderemo due condensatori da 100 pF (numerazione 101) in C1 e C2.

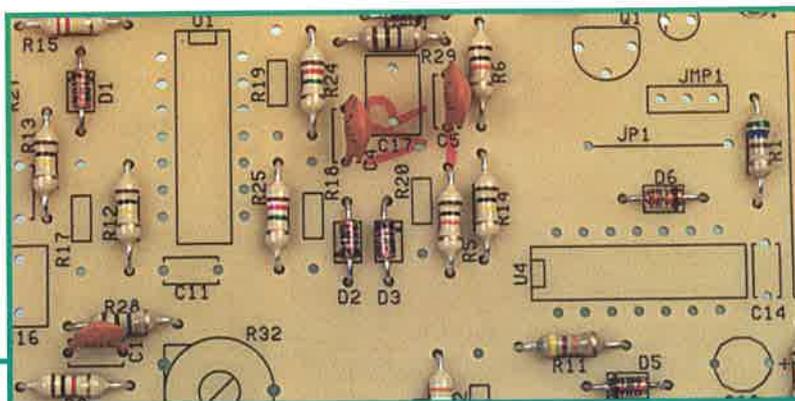
Salderemo altri tre condensatori ceramici, il cui valore è di 10 nF, e vanno collocati in C3, C4 e C5. Il verso di inserzione di questi condensatori non è importante, li riconosceremo dalla scritta 103Z che portano impressa.



Due condensatori ceramici, da 22 nF di valore, saranno montati in C6 e C7. Portano scritto il numero 223Z e sono di dimensione maggiore rispetto ai condensatori che abbiamo saldato in precedenza.



Infine salderemo 6 diodi di segnale 1N4148, che saranno montati in D1, D2, D3, D4, D5 e D6. A differenza del resto dei componenti che abbiamo montato sino ad ora sulla scheda dei sensori, i diodi hanno polarità. Dobbiamo far coincidere la banda nera che ogni diodo possiede con la linea nera disegnata sulla serigrafia della scheda.



```

c:\progra~1\mplab\faso_36\esempio1.asm
1
2
3      List      p=16F84
4      include "P16F84.INC"
5
6      Numero    equ      0x0c
7      Delay_Cont equ      0x0d
8      Temporaneo equ      0x0e
9
10     org      0x00
11     goto     Inizio
12
13
14     Tabella:   addwf     PCL,F
15     retlw    b'00111111'
16     retlw    b'00000110'
17     retlw    b'01011011'
18     retlw    b'01001111'
19     retlw    b'01100110'
20     retlw    b'01101101'
21     retlw    b'01111101'
22
23

```

L'esercizio di cui ci occupiamo è un po' più complesso di quelli precedenti. Si tratta di implementare un dado elettronico, e il suo enunciato è il seguente: dobbiamo generare un numero casuale fra 1 e 6. Quando RA0 sarà a "1" sul display a 7 segmenti collegato alla porta B si vedranno sequenzialmente i numeri da 1 a 6, con intervalli di 0,05. Al momento del passaggio di RA0 a "0" si vedrà, per 3 secondi, il numero casuale ottenuto. Dopo il display si spegne e la sequenza ricomincia. Nell'immagine abbiamo la tabella della routine. Essa converte il codice binario presente nei quattro bit meno significativi del registro W nell'equivalente a 7 segmenti. Il codice per i 7 segmenti viene scritto anch'esso in W.

Delay_20_ms: questa routine di temporizzazione ha come obiettivo l'eliminazione dell' "effetto rimbalzo" delle periferiche elettromeccaniche. Realizza un ritardo di 20 ms. Se il PIC lavora a una frequenza di 4 MHz, il TMR0 si incrementa ogni microsecondo. Se vogliamo temporizzare 20.000 µs (20 ms) con un prescaler da 256, il TMR0 dovrà contare 78 eventi (78 * 256). Il valore 78 equivale a 0x4e hex, e dato che il TMR0 è ascendente dovremo caricarlo con il suo complemento a 1 (0xb1 hex). *Dealy_var:* routine simile a quella precedente. Il TMR0 realizza una temporizzazione di 50 ms che si ripete tante volte quante indica la variabile *Delay_Cont*. Si ottengono temporizzazioni fra 50 ms e 12,6 sec.

```

c:\progra~1\mplab\faso_36\esempio1.asm
24     Delay_20_ms:  bcf      INTCON,T0IF
25     movlw        0xb1
26     movwf        TMR0
27     Delay_20_ms_1  clrwdt
28     btfss        INTCON,T0IF
29     goto         Delay_20_ms_1
30     return
31
32
33
34     Delay_var:     bcf      INTCON,T0IF
35     movlw        0x3c
36     movwf        TMR0
37     Intervallo   clrwdt
38     btfss        INTCON,T0IF
39     goto         Intervallo
40     decfsz       Delay_Cont,F
41     goto         Delay_var
42     return
43

```

```

c:\progra~1\mplab\faso_36\esempio1.asm
44     Inizio        clr     PORTB
45                    bsf     STATUS,RP0
46                    clr     TRISB
47                    movlw   b'00011111'
48                    movwf   TRISA
49                    movlw   b'00000111'
50                    movwf   OPTION_REG
51                    bcf     STATUS,RP0
52
53     Ciclo          clrwdt
54                    btfss   PORTA,0
55                    goto    Ciclo
56                    movf    TMR0,W
57                    movwf   Numero
58                    call    Delay_20_ms
59
60
61     Dividi:        movlw   d'6'
62                    subwf   Numero,W
63                    movwf   Numero
64                    sublw   d'5'
65                    btfss   STATUS,C
66                    goto    Dividi
67                    incf   Numero,F

```

Dopo la classica configurazione delle porte, si arriva alla parte principale del programma che consiste nel rilevare l'azionamento di RA0, evitando i rimbalzi che si potrebbero produrre. Dopo questo, si legge il valore attuale del TMR0, che rappresenterà la base del nostro numero casuale, e si divide per 6 ottenendo un numero casuale compreso fra 1 e 6, corrispondente alle 6 facce del dado.

```

c:\progra~1\mplab\ fasc_36\esempio1.asm
68
69 Dado:      movlw   d'6'
70           movwf   Temporaneo
71 RA0_1     clrwdt
72           btfss   PORTA,0
73           goto    Uscita
74           movf   Temporaneo,W
75           call   Tabella
76           movwf  PORTB
77           movlw  d'1'
78           movwf  Delay_Cont
79           call   Delay_var
80           decfsz Temporaneo,F
81           goto  RA0_1
82           goto  Dado
83
84           call   Delay_20_ms

```

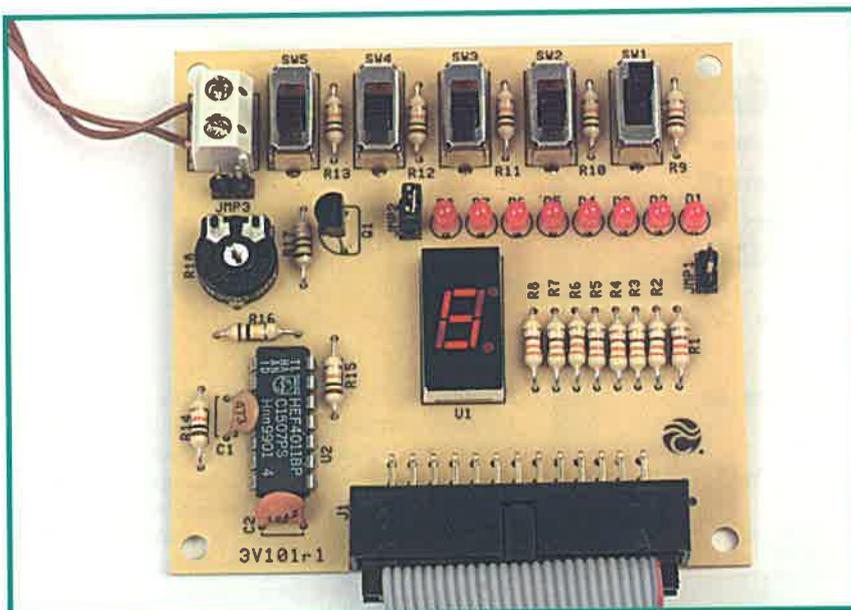
Questa sequenza di istruzioni ha come scopo di visualizzare sul display i numeri da 1 a 6 a intervalli di 0,05s per dare la sensazione di movimento del dado. Questo movimento si mantiene sino a quando RA0 è a "1". Mettendo RA0 a "0" si visualizza il numero casuale catturato in precedenza dal TMRO.

La routine uscita serve per scegliere un numero casuale e visualizzarlo sul display. Inoltre carica il valore dei temporizzatori in modo che torni a eseguirsi il ciclo del programma. Osservate che la routine di temporizzazione chiamata durante il programma, ha una doppia funzione: generare i numeri casuali ed eliminare i rimbalzi.

```

c:\progra~1\mplab\ fasc_36\esempio1.asm
85
86 Uscita:   movf   Numero,W
87           call   Tabella
88           movwf  PORTB
89           movlw  d'60'
90           movwf  Delay_Cont
91           call   Delay_var
92           clrf   PORTB
93
94           goto  Ciclo
95
96           end
97

```

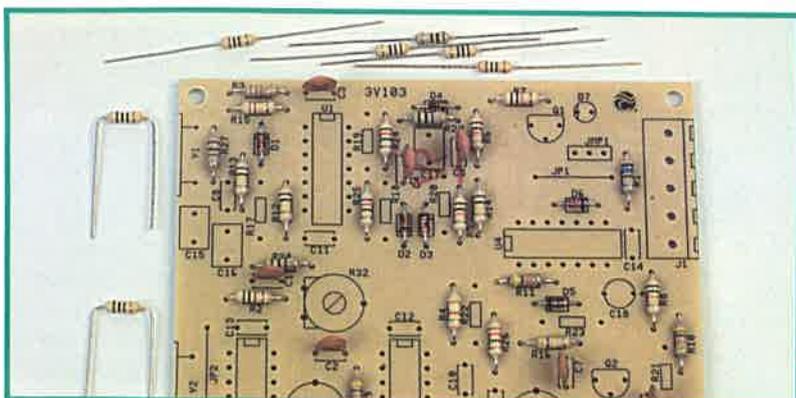


Per provare questo programma scriveremo il PIC tramite la scheda di controllo, e in seguito la collegheremo a quella di ingressi e uscite. È necessario lasciare aperti tutti i jumpers eccetto JMP2, dato che utilizzeremo il display a 7 segmenti per visualizzare il numero del dado.

MODULO DEI SENSORI

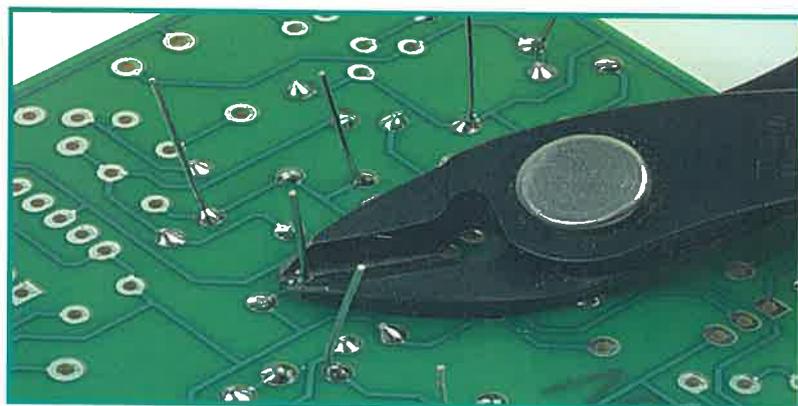
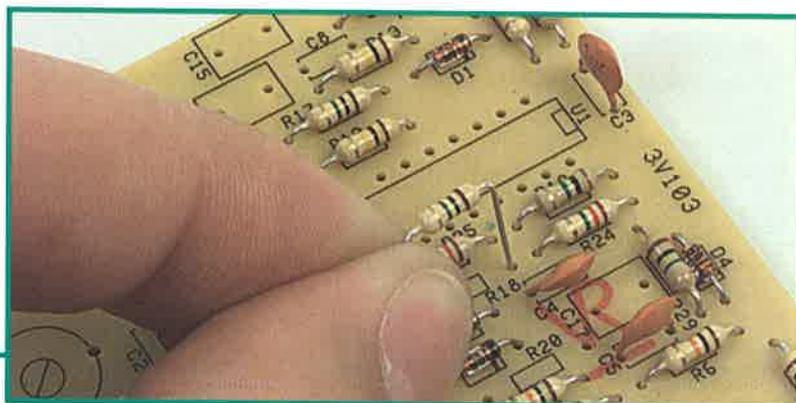
PL 72

Montaggio passo a passo



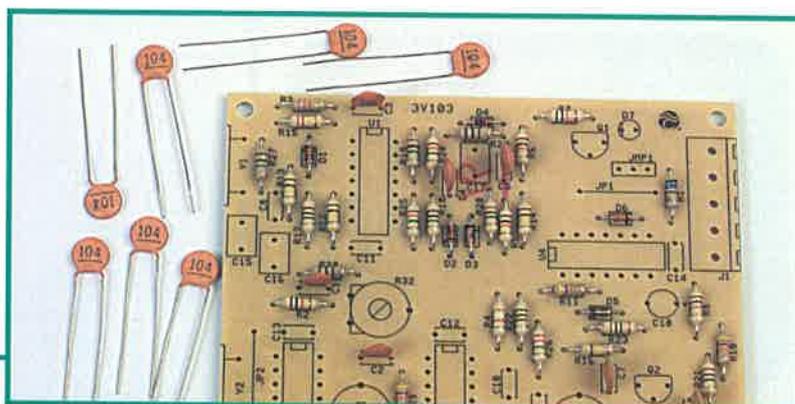
Salderemo sette resistenze del valore di 1 M Ω . Il loro codice a colori è marrone-nero-verde, e dovranno essere montate in R17, R18, R19, R20, R21, R22 e R23.

Dobbiamo inserirle sino a che siano ben appoggiate sulla scheda. Anche se non hanno polarità, dovremo orientarle tutte con la banda dorata rivolta nel medesimo verso.

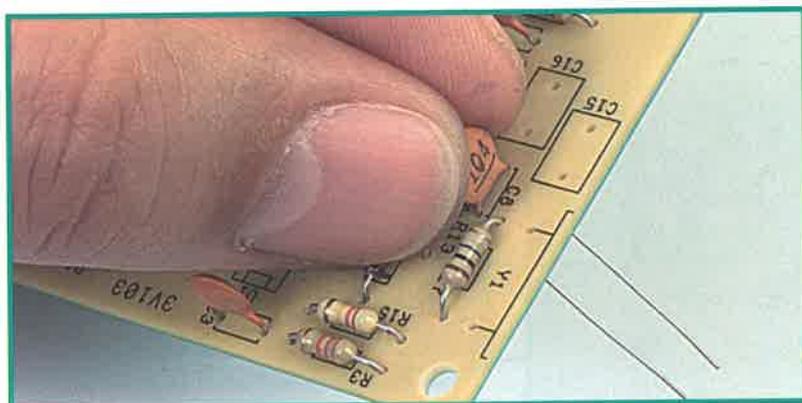


Dopo aver realizzato le saldature, è necessario tagliare la parte dei reofori in eccedenza con un tronchesino come quello che vedete nell'immagine. Le saldature, come sempre, dovranno essere a forma di cono e brillanti.

In seguito salderemo sette condensatori ceramici da 100 nF. Come gli altri condensatori ceramici che abbiamo già saldato, non hanno polarità, quindi il verso di inserzione è indifferente.

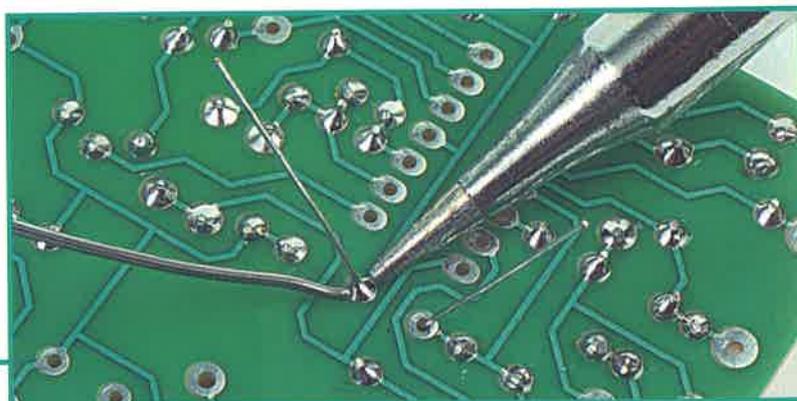


Montaggio passo a passo

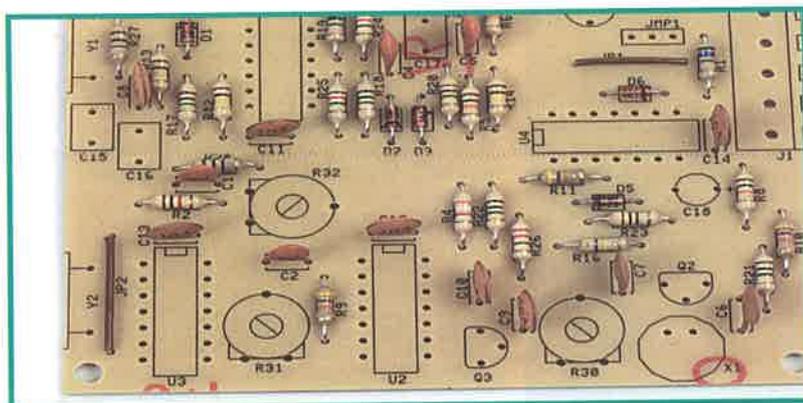


Monteremo i condensatori nelle posizioni C8, C9, C10, C11, C12, C13 e C14. Li inseriremo sino a che non restino bene appoggiati alla scheda, come mostrato nell'immagine.

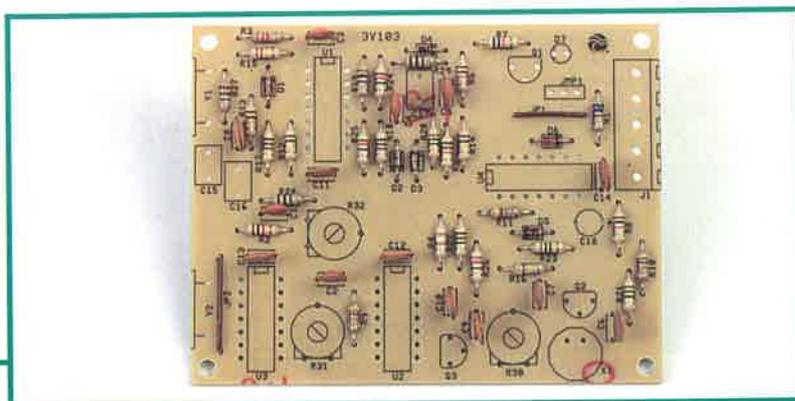
Saldando un condensatore ceramico, dobbiamo realizzare la saldatura più rapidamente di quando saldiamo, ad esempio, una resistenza. Se riscaldiamo il condensatore troppo a lungo con il saldatore, la capsula potrebbe danneggiarsi.

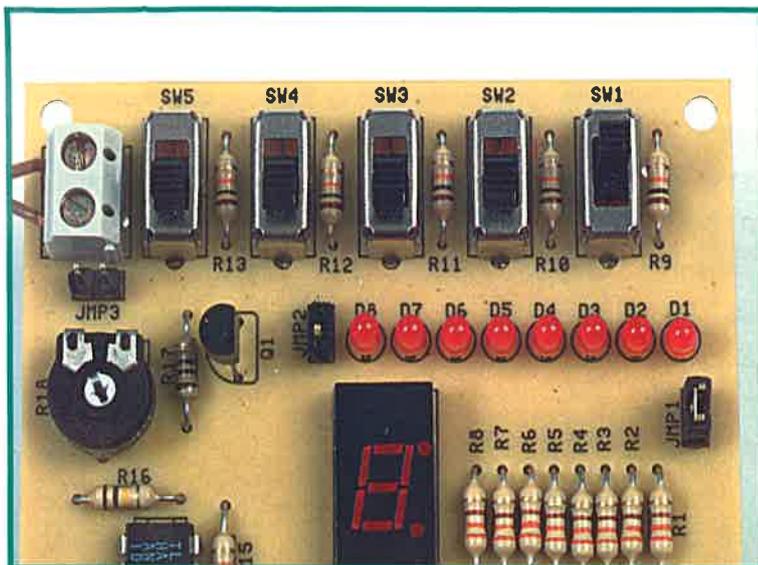


Montiamo ora i ponticelli JP1 e JP2 della scheda. Per fare questo taglieremo due pezzi di cavo della dimensione di ogni ponticello e ne pelereemo gli estremi. Li introdurremo come mostrato nell'immagine, quindi li salderemo secondo la solita procedura.



Ecco come deve rimanere la scheda dei sensori arrivati a questo punto. Abbiamo saldato tutte le resistenze, i diodi, la gran parte dei condensatori, e anche i due ponticelli. Si raccomanda di verificare la corretta inserzione dei componenti, così come le saldature dei medesimi.





Realizziamo ora un esercizio con i diodi della porta B, con cui otterremo un estetico movimento: faremo una rotazione sequenziale nell'accensione di ogni LED collegato alla porta B. Se $RA0 = 0$, la rotazione sarà da destra a sinistra e viceversa. Ogni LED rimarrà acceso 0,25 secondi (250 ms). L'immagine mostra la disposizione dei LED sulla scheda di ingressi e uscite.

Come già sappiamo, prima di tutto selezioneremo il tipo di dispositivo e includeremo il file con le definizioni del registro. Nomineremo un registro di utilizzo generale con il nome di contatore, a partire dalla prima posizione libera, che è la 0x0c. Il reset lo collocheremo nel vector 0 come sempre, e salteremo il vector di interrupt, quindi il programma inizierà alla posizione 5.

```

c:\progra~1\mplab\asc_37\esempio1.asm
1
2
3
4
5 Contatore      equ    0x0c
6
7
8
9
10
11
List    p=16F84
include "P16F84a.INC
    
```

```

c:\progra~1\mplab\asc_37\esempio1.asm
12 *****
13 ;Routine di ritardo di 250 ms
14
15 Delay          movlw   .10
16               movwf   Contatore
17 Delay_0        bcf     INTCON,T0IF
18               movlw   0x3c
19               movwf   TMR0
20 Delay_1        clrwdt
21               btfss   INTCON,T0IF
22               goto    Delay_1
23               decfsz  Contatore,F
24               goto    Delay_0
25               return
26
    
```

Questa è la routine di temporizzazione o Delay, che utilizzeremo in questo programma. Se il PIC lavora ad una frequenza di 4 MHz, il TMR0 cambia ogni microsecondo. Se si desidera temporizzare 25.000 microsecondi (25 ms) con un prescaler di 128, il TMR0 dovrà contare 195 eventi ($195 * 128 = 24.960$). Il valore 195 equivale a 0xc3 e, dato che il TMR0 è ascendente dovremo caricare il suo complemento (0x3c). Questa temporizzazione si ripeterà 10 volte per ottenere il totale desiderato di 250 ms.

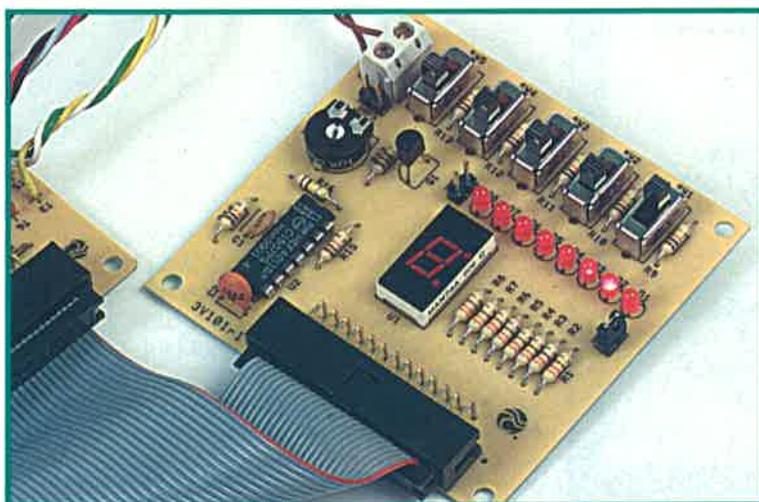
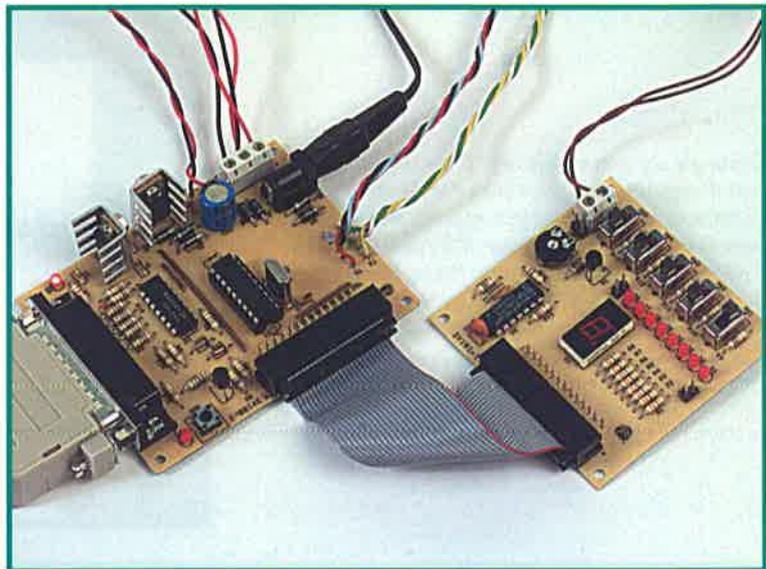
```

c:\progra~1\mplab\masc_37\esempio1.asm
27 *****
28 ;Inizio del programma principale
29
30 Inizio      clr    PORTB
31            bsf    STATUS,RP0
32            clr    TRISB
33            movlw b'00011111'
34            movwf TRISA
35            movlw b'00000110'
36            movwf OPTION_REG
37            bcf    STATUS,RP0
38
39            bsf    STATUS,C
40 Ciclo      call   Delay
41            btfsc PORTA,0
42            goto  A_Destra
43 A_Sinistra rlf    PORTB,F
44            goto  Ciclo
45 A_Destra   rrf    PORTB,F
46            goto  Ciclo
47
48            END

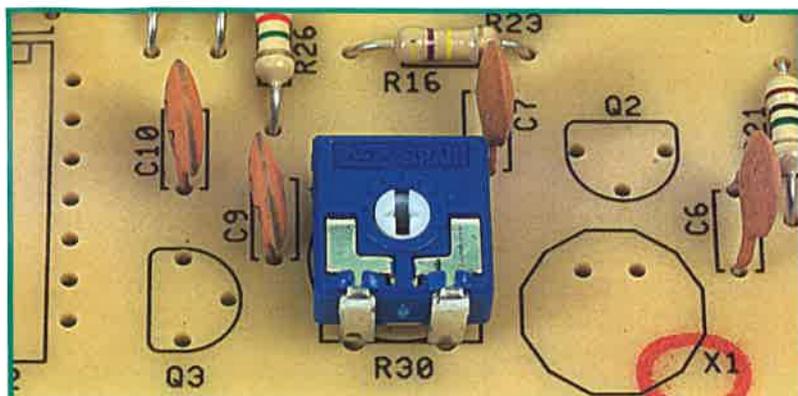
```

All'inizio del programma configureremo la porta B come uscita per i LED e la porta A come ingresso. Assegneremo un prescaler di 128 al Timer0, per fare in modo che la nostra routine di temporizzazione funzioni correttamente. Ora faremo ruotare l'accensione dei LED verso sinistra o verso destra, in funzione dello stato dell'interruttore in RA0. Per realizzare le rotazioni useremo le istruzioni di rotazione di registro: rrf e rlf.

Dopo aver scritto e compilato il programma procederemo alla sua memorizzazione mediante il software del PC e la scheda di controllo. Ricordate che dovete sempre disabilitare i jumper della scheda I/O durante la memorizzazione, nel caso in cui questa si trovi unita mediante il Pic Bus alla scheda di controllo, e che bisogna collocare l'interruttore di scrittura della scheda di controllo in posizione PROG e non in RUN. Una volta scritto il PIC sposteremo nuovamente l'interruttore su RUN.

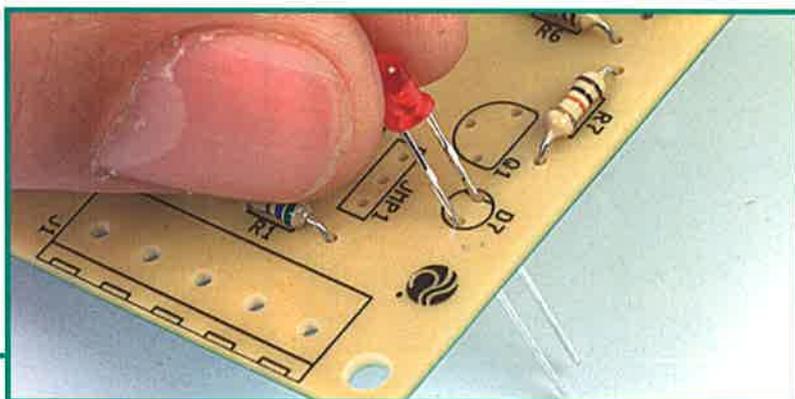


Per verificare il funzionamento di questo programma dobbiamo montare il jumper JMP1 che attiva i diodi LED: il resto dei jumper devono essere aperti. Vedremo che modificando lo stato di RA0 otterremo che il "gioco di luci" ruota in un verso o nell'altro. Proponiamo al lettore di realizzare un esercizio simile, però con la temporizzazione di 100 o 250 ms fra l'accensione di un LED e il successivo. Queste temporizzazioni si selezionano mediante SW1 (RA0). Con l'interruttore RA2 si controlla il verso dello spostamento.

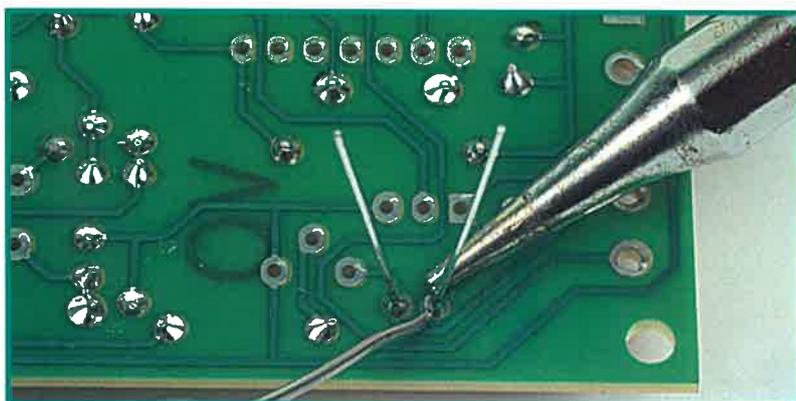


Monteremo il potenziometro da 4K7 Ω in R30 sino a che sia totalmente inserito, e procederemo alla sua saldatura. Ha un solo orientamento possibile.

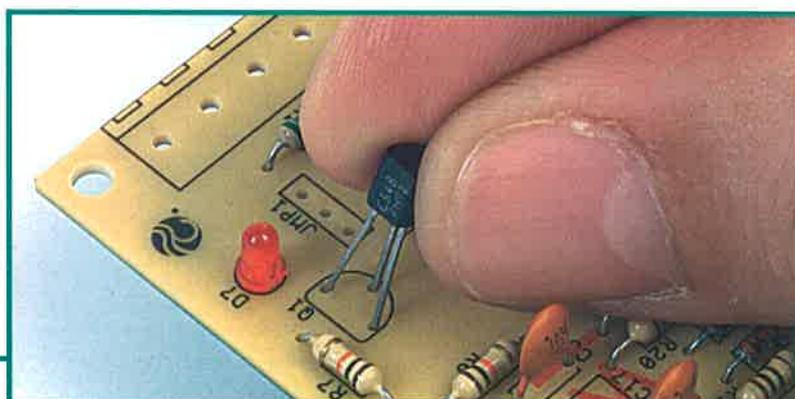
Dobbiamo inserire il diodo LED da 3 mm in D7. Questo diodo ha polarità, che è segnata sulla serigrafia della scheda con una doppia linea. Dobbiamo far coincidere questo riferimento della scheda con il pin più corto del diodo, il catodo.



Una volta inserito il LED, procederemo alla sua saldatura verificando che sia ben introdotto e che sia dritto. Inoltre dobbiamo verificare la corretta polarità prima di tagliare i piedini del LED.



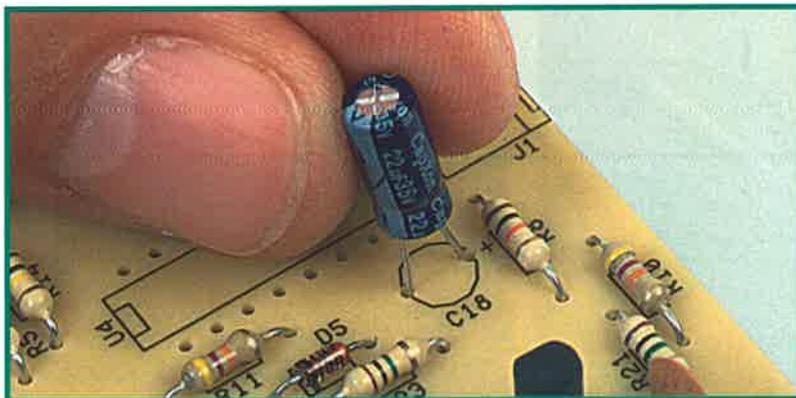
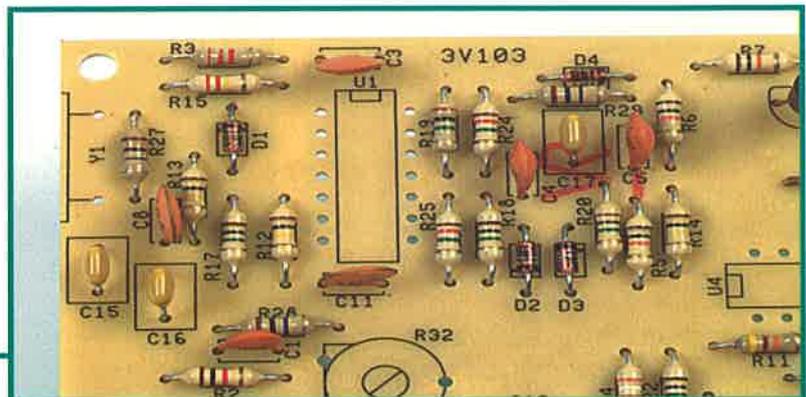
Montiamo ora il transistor Q1. Inseriremo un transistor modello BC547, il quale non deve essere completamente inserito sulla scheda, ma lasciato sollevato sulla superficie di qualche millimetro.





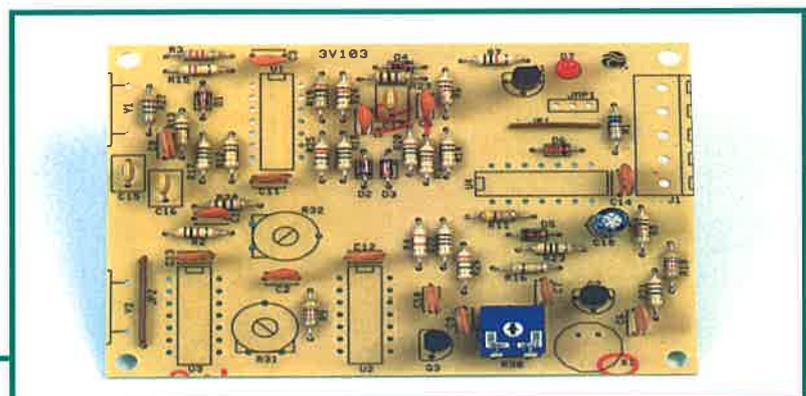
Allo stesso modo di Q1 installeremo altri due transistor BC547 in Q2 e Q3. Dobbiamo inserire tutti i transistor in modo che la loro vista superiore faccia coincidere la forma a mezzaluna del transistor con la forma a mezzaluna serigrafata sulla scheda. Salderemo i transistor.

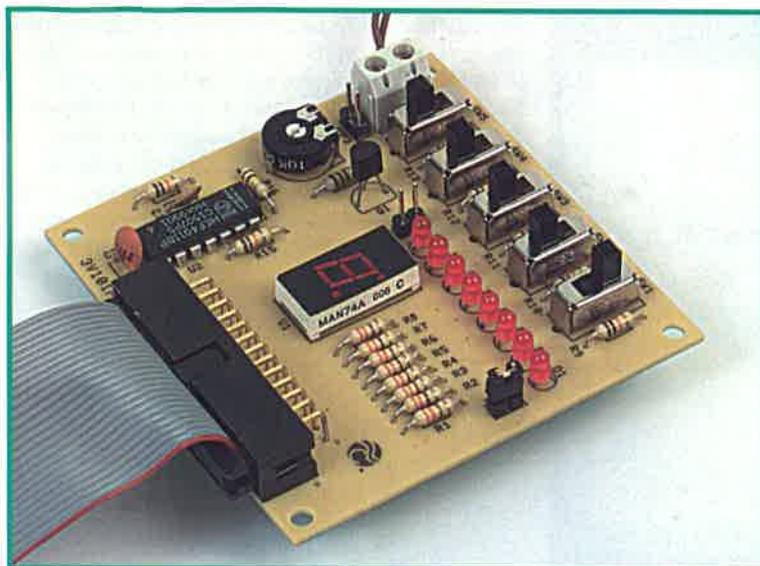
Monteremo e salderemo tre condensatori ceramici da 470 nF. Le loro posizioni sulla scheda sono C15, C16 e C17. Questi condensatori non hanno polarità, quindi è indifferente il loro verso di inserzione.



Ora montiamo un condensatore elettrolitico da 22 μ F/35 V, il quale ha polarità, in C18. Nella serigrafia della scheda viene indicato con il simbolo + il posto dove dovremo montare la parte positiva del condensatore. La polarità di questo condensatore viene indicata sul suo contenitore, dove è segnalato il piedino che corrisponde all'armatura negativa.

Questo è l'aspetto della scheda dei sensori in questo momento. Come sempre, dobbiamo verificare di aver saldato tutti i componenti al loro posto, e quelli che hanno polarità debbono essere disposti nel verso appropriato.





Realizzeremo ora un altro esercizio sequenziale, questa volta si tratta di una macchina di riempimento. Due relè, "M1" (RB0), e "M2" (RB1), pilotano due motori che muovono due nastri trasportatori. "M1" (RB0) trasporta pezzi e "M2" (RB1) contenitori. Un sensore "DP" (RA1) rileva il passaggio dei pezzi, e un altro "DE" (RA2), il corretto posizionamento dei contenitori. Quando sono passati 10 pezzi il contenitore si considera pieno, si attiva un segnale acustico "A" (RB2) e il nastro che trasporta gli imballaggi si sposta sino a presentare un contenitore vuoto. In questo momento si disattiva il segnale acustico "A" (RB2) e avanza nuovamente il nastro dei pezzi ripetendo così il ciclo. L'interruttore "I" (RA0) attiva o disattiva il sistema. L'immagine mostra la disposizione degli interruttori che simulano i sensori, così come i LED che simulano i relè.

Selezioniamo il tipo di dispositivo, includiamo il file con la definizione dei registri, e nominiamo mediante un'etichetta il primo registro libero di utilizzo generale. Saltiamo il vector di interrupt, e iniziamo lo sviluppo del programma. La prima cosa da fare sarà configurare la porta B come uscita e la porta A come ingresso per la rilevazione dei sensori. Per fare questo dobbiamo passare al Banco 1 come abbiamo fatto in tutti gli esercizi.

```

c:\progra~1\mplab\masc_38\esempio1.asm
1
2
3
4
5 Conta_pezzi
6
7
8
9
10
11 INIZIO
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

List p=16F84a
include "P16F84a.INC"

Conta_pezzi equ 0x0c

org 0x00
goto INIZIO
org 0x05

INIZIO
    clrf PORTB
    bsf STATUS,RP0
    clrf TRISB
    movlw b'00011111'
    movwf TRISA
    bcf STATUS,RP0
    
```

```

c:\progra~1\mplab\masc_38\esempio1.asm
18 ;*****
19
20 Loop_0
21 Loop:
22     clrf PORTB
23     clrwdt
24     movlw d'10'
25     movwf Conta_pezzi
26     btfss PORTA,0
27     goto Loop_0
28
29     bcf PORTB,0
30     bsf PORTB,1
31
32     clrdt
33     btfss PORTA,2
34     goto No_vaso
35
36     bsf PORTB,0
37     bcf PORTB,1
38     bcf PORTB,2
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
    
```

Dopo aver scollegato le uscite, la prima cosa che faremo sarà inizializzare il numero di pezzi che dobbiamo contare. Attenderemo che il sensore attivi il sistema mediante RA0 (I), quindi metteremo in marcia il nastro trasportatore di contenitori, sino a che il sensore di rilevamento su RA2 non ne indichi il corretto posizionamento. Quando questo avverrà fermeremo il nastro trasportatore dei contenitori e attiveremo il nastro dei pezzi.

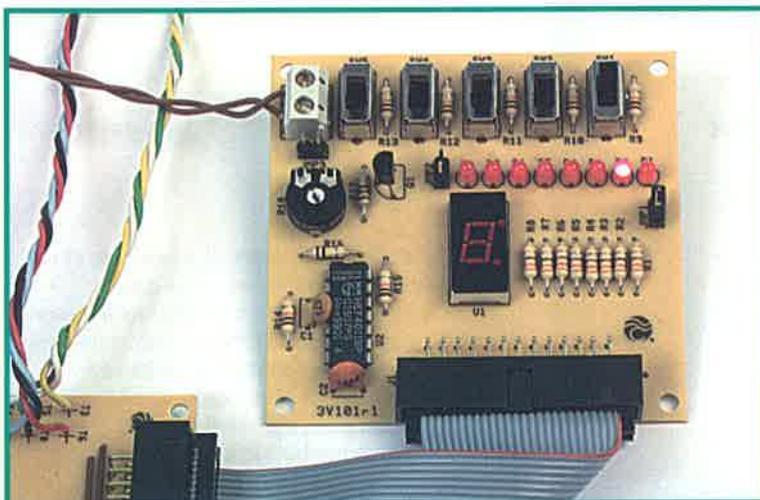
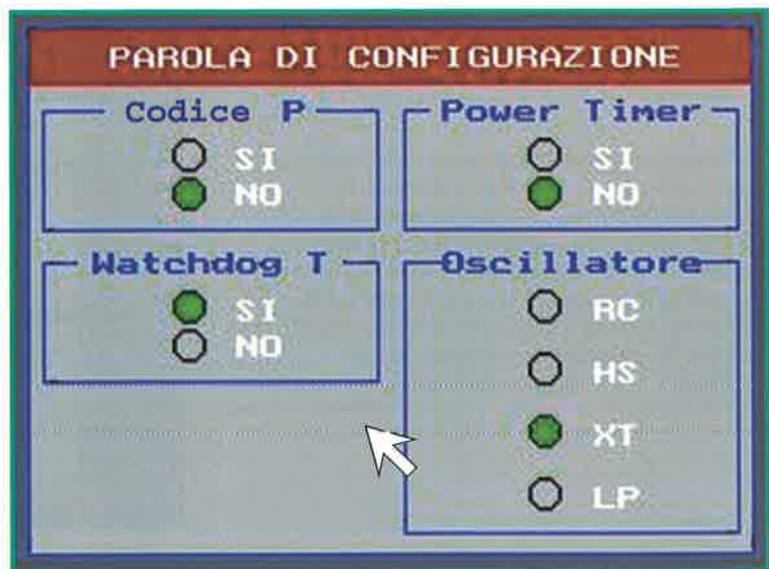
```

c:\progra~1\mplab\masc_38\esempio1.asm
41 Pezzo_0      clrwdt
42             btfss   PORTA,1
43             goto   Pezzo_0
44
45 Pezzo_1      clrwdt
46             btfsc   PORTA,1
47             goto   Pezzo_1
48             decfsz  Conta_pezzi,F
49             goto   Pezzo_0
50             bcf    PORTB,0
51             bsf    PORTB,1
52             bsf    PORTB,2
53
54 Si_vaso     clrwdt
55             btfsc   PORTA,2
56             goto   Si_vaso
57
58             goto   Loop
59
60             end

```

Ora aspettiamo che passino i dieci pezzi, operazione che realizzeremo con il sensore di posizionamento in RA1. Dovremo rilevare sia l'ingresso dei pezzi sia la loro uscita. Conteremo sino a 10, fermeremo il motore del nastro dei pezzi e attiveremo quello dei contenitori. Inoltre verificheremo che arrivi un nuovo contenitore sino al sensore di posizionamento, controllando che il contenitore precedente sia uscito.

Dopo aver scritto il programma procederemo alla sua memorizzazione. Questo programma è stato preparato per funzionare con il Watchdog attivato. Potremo scriverlo, pertanto, attivando questa opzione della parola di configurazione del PIC. Con il Watchdog su ON ci assicureremo che se il programma cade in un ciclo infinito, si resett automaticamente per sbloccarsi. Osservate che per rendere possibile questo, è stato necessario introdurre le corrispondenti istruzioni `clrwdt` lungo il programma.

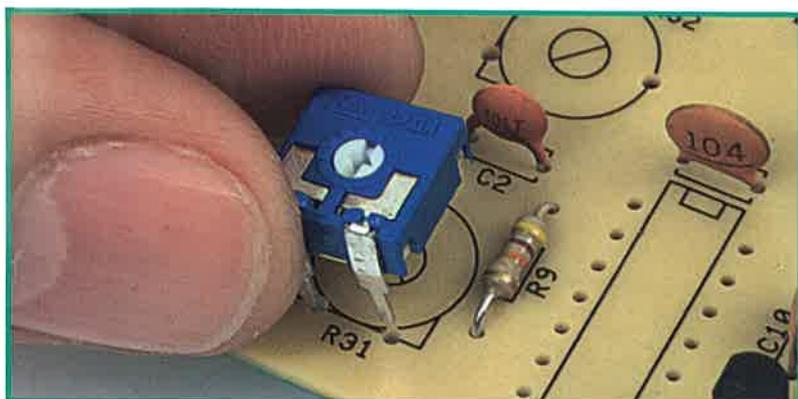


Dopo aver programmato il PIC e collegato il Pic Bus sulla scheda di I/O e quella di controllo, chiuderemo il jumper JMP1 per poter vedere l'accensione dei LED. Ora mediante gli interruttori simuleremo l'attivazione e la disattivazione dei sensori, con cui potremo provare il programma. Sugeriamo al lettore di fare pratica introducendo nuovi sensori e nastri trasportatori all'enunciato di questo esempio, dato che è importante fare pratica con programmi di tipo sequenziale, utili per la successiva programmazione di Monty.

MODULO DEI SENSORI

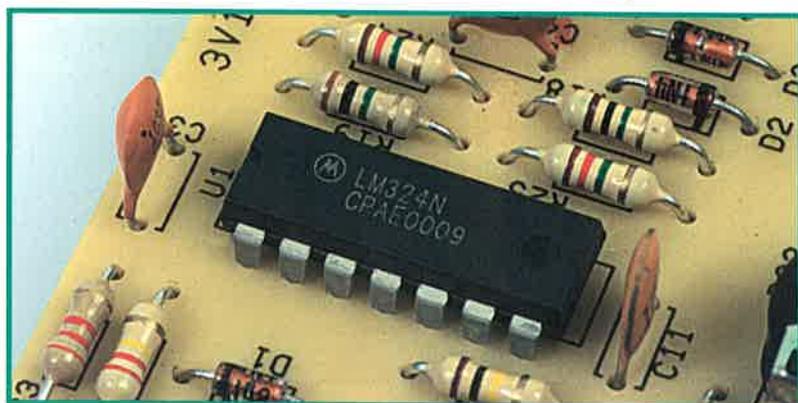
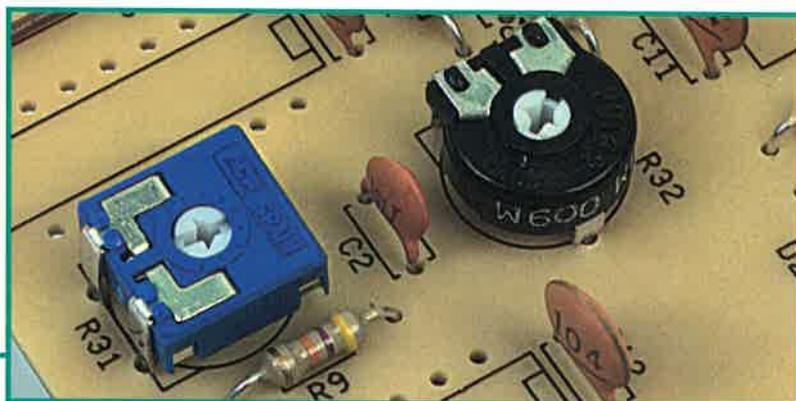
PL 76

Montaggio passo a passo



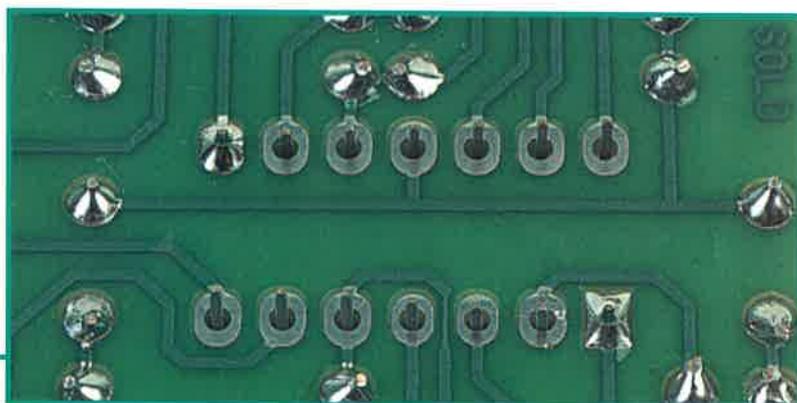
Per prima cosa inseriremo e salderemo un potenziometro da 10 K Ω . Lo collocheremo in R31, e lo posizioneremo a filo della scheda, quindi procederemo alla saldatura.

Continueremo saldando un altro potenziometro in R32. Monteremo qui il potenziometro da 1 M Ω . Come il precedente andrà inserito a filo della scheda, e poi saldato.

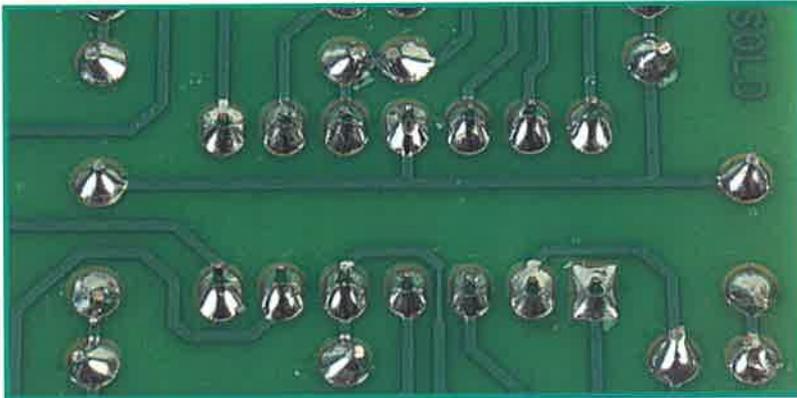


Ora dobbiamo saldare il circuito integrato LM324 in U1. È fondamentale inserire il circuito integrato in modo corretto, facendo coincidere la tacca di riferimento presente sull'integrato con quella presente sulla serigrafia. Deve rimanere come mostrato nell'immagine.

Per prima cosa salderemo solo i due pin opposti del circuito integrato, verificheremo quindi il corretto verso di inserzione e che sia a filo della scheda.

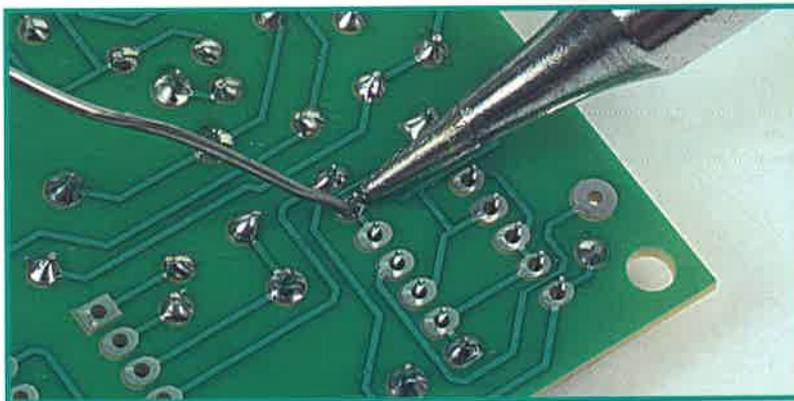
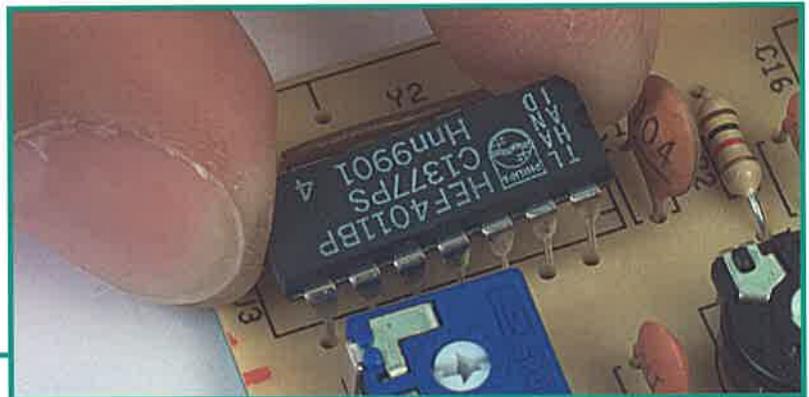


Montaggio passo a passo



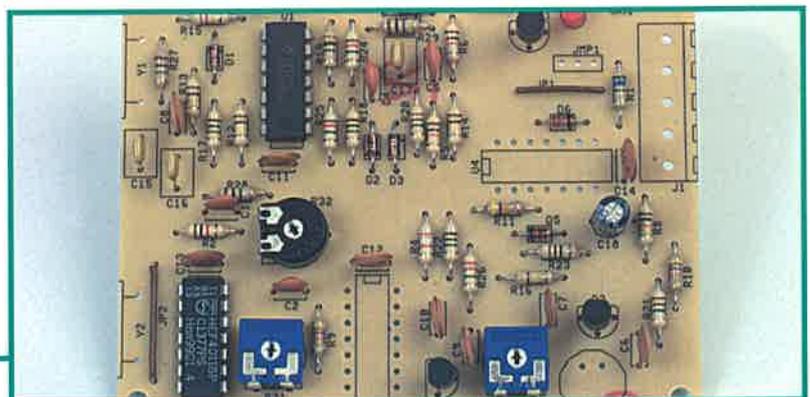
Quando saremo sicuri che il circuito integrato sia stato montato correttamente sulla scheda dei sensori, salderemo il resto dei piedini. Le saldature debbono rimanere come quelle dell'immagine, a forma di cono e ben appoggiate sulla superficie. Fate attenzione a non riscaldare in modo eccessivo il componente.

Dobbiamo montare un altro circuito integrato, in questo caso sarà il circuito 4011, il quale va su U3, nella parte inferiore sinistra della scheda. Così come per l'LM324, inseriremo il chip nel verso adeguato, facendo coincidere la tacca di riferimento con quella disegnata sulla serigrafia.



Per saldare il circuito seguiremo lo stesso metodo utilizzato per l'integrato U1. Salderemo i due estremi, ci assicureremo del corretto verso di inserzione e procederemo a saldare il resto dei piedini.

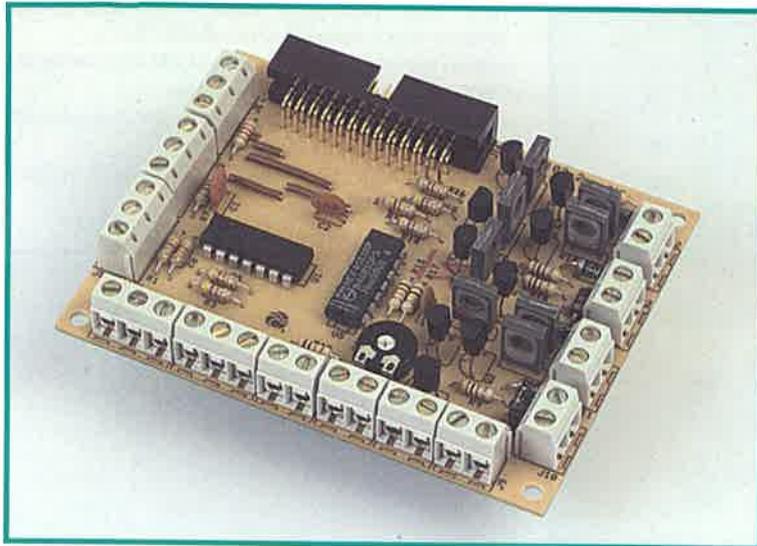
Aspetto della scheda dei sensori con tutti i componenti saldati sino a questo momento. Dobbiamo verificare tutte le parti montate, specialmente quelle dotate di polarità, dato che se fossero male inserite, al momento dell'accensione si potrebbe danneggiare la scheda.



MODULO DI POTENZA

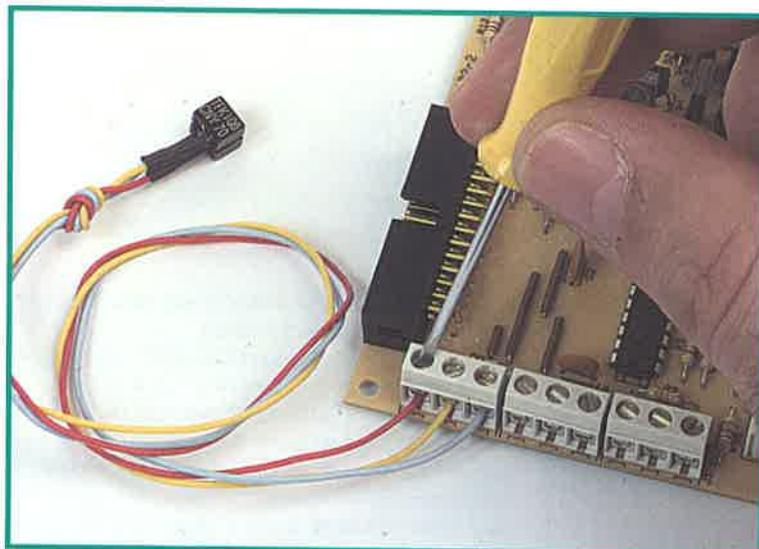
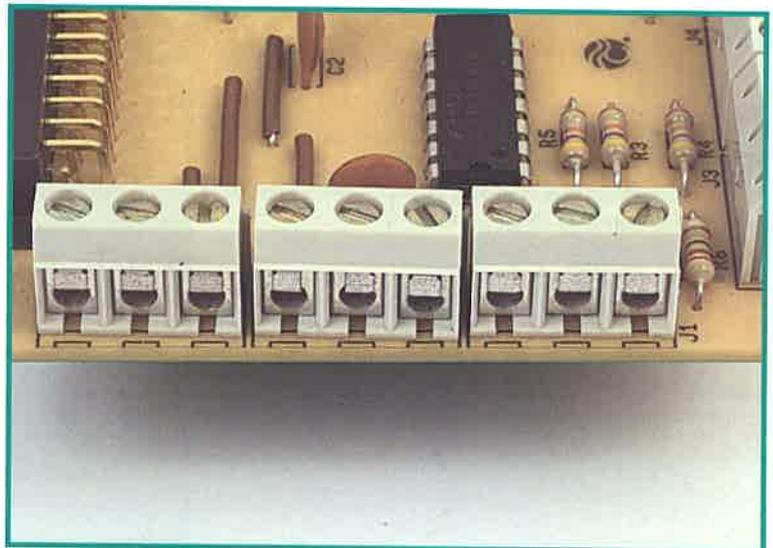
PL 77

Consigli pratici. Utilizzo dei sensori

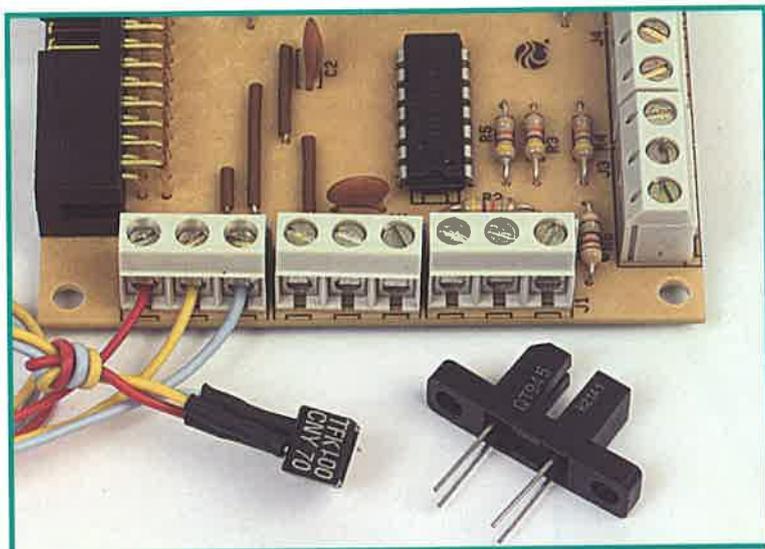


Descriveremo tutte le morsettiere di cui è composta la scheda di potenza, per collegare ad essa i sensori ottici e meccanici di cui disponiamo, e realizzare esercizi con i sensori reali che possiede il robot. Ricordate che a questa scheda inoltre si collegheranno i motori che ricevono il controllo e l'alimentazione tramite i ponti ad H formati dai diversi transistor.

Le morsettiere J1, J2 e J5 hanno tre terminali e sono predisposte per il collegamento dei sensori ottici. Se le guardiamo di fronte, nel terminale di sinistra di ognuna di esse abbiamo il +5V, in quello al centro l'ingresso per il fototransistor (troviamo la resistenza da 47 K e il trigger 4016), e nel terminale di destra l'ingresso per il fotodiode (resistenza da 220 Ω collegata a massa).

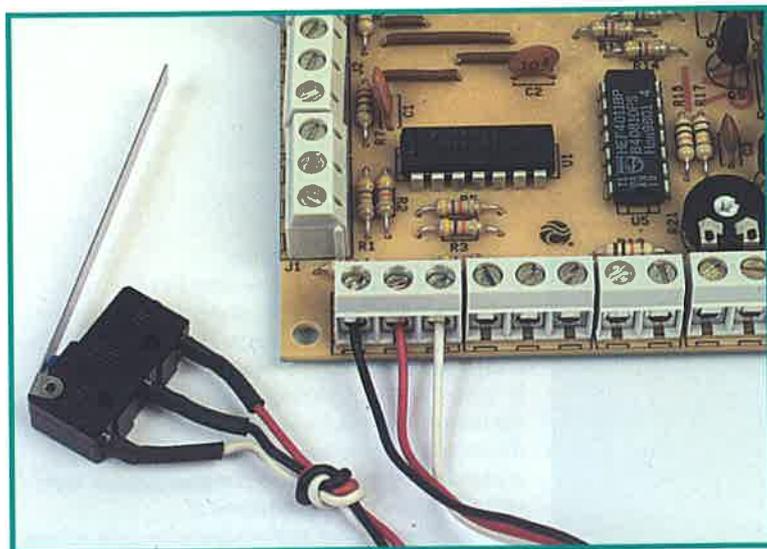
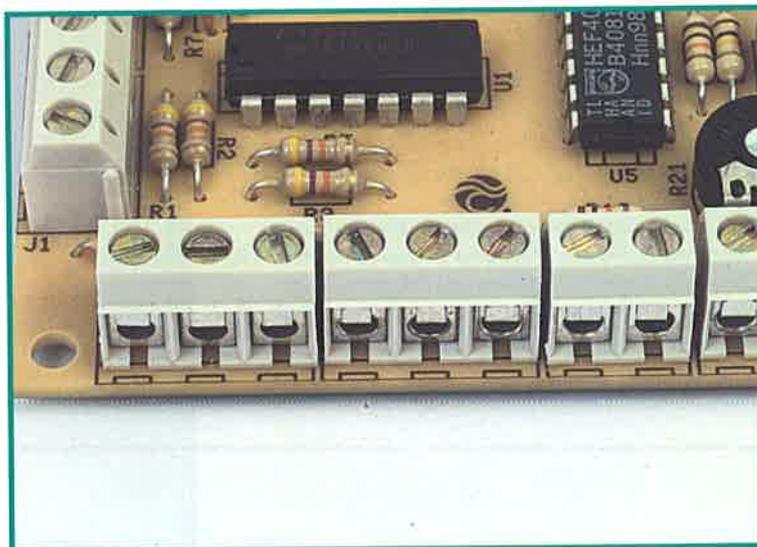


In questo modo per collegare un sensore CNY70, collegheremo il cavo su cui abbiamo unito i due terminali del sensore sul morsetto di sinistra del +5 V. Inseriremo il cavo del piedino che si trova sotto la sigla CNY70 sul morsetto centrale (uscita del fototransistor), e l'ultimo filo sul morsetto di destra (fotodiode). In questo modo rimane alimentato il sensore ottico che invia segnali logici verso il PIC grazie alla circuiteria di condizionamento del segnale presente sulla scheda.



Nello stesso modo in cui abbiamo collegato questo sensore ottico collegheremo gli altri due, uno in ogni morsettiere. Oltre al sensore a riflessione CNY70, potremo approfittare della circuiteria della scheda di potenza per collegare ad essa qualsiasi tipo di sensore ottico, come il sensore a forcella che si vede nell'immagine. Il procedimento per il collegamento è esattamente uguale.

Le morsettiere J3 e J4 servono per il collegamento dei sensori di tipo meccanico, in queste morsettiere non possiamo collegare i sensori ottici. Normalmente utilizzeremo queste morsettiere per collegare i finecorsa che ci serviranno per rilevare gli ostacoli e per evitare che Monty entri in collisione. Se osserviamo le morsettiere di fronte, nel terminale di sinistra abbiamo la massa, in quello al centro +5 V e in quello a destra l'ingresso per il segnale del sensore.



Qui abbiamo un sensore meccanico tipo finecorsa, collegato alla scheda di potenza. Il primo piedino del finecorsa è l'uscita del sensore che collegheremo con il terminale di destra. Poi collegheremo gli altri due terminali del sensore negli altri ingressi della morsettiere. In funzione di come realizzeremo il collegamento, il nostro sensore meccanico sarà attivo con livello basso o alto.

MODULO DEI SENSORI

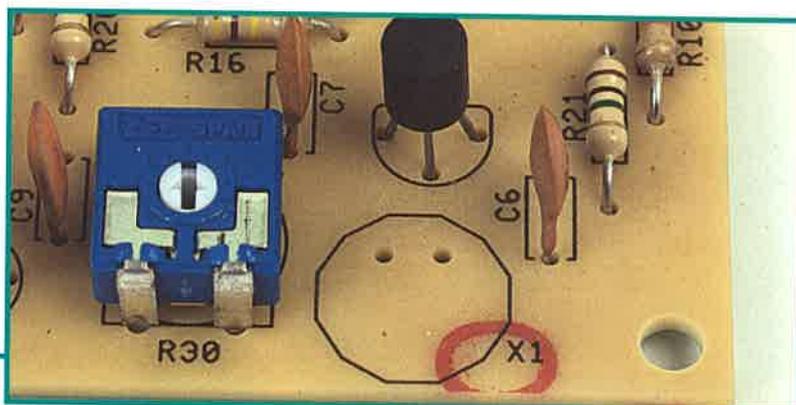
PL 78

Montaggio passo a passo



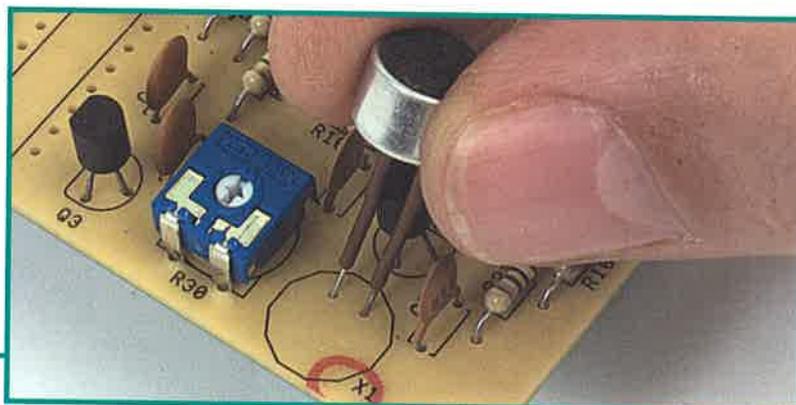
Dobbiamo saldare un microfono alla scheda dei sensori, che ci servirà come sensore per la rilevazione dei suoni.

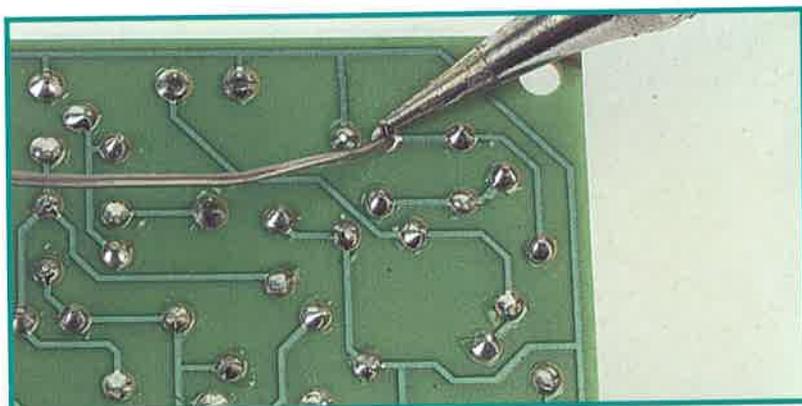
Prima di montare il microfono sulla scheda dei sensori dobbiamo identificare la sua posizione all'interno della scheda stessa, dove sarà inserito nella posizione X1. Il microfono non ha polarità.



Per montarlo dobbiamo saldare due fili rigidi sui terminali posti sul contenitore del microfono. Utilizzeremo il filo marrone che ci è avanzato da un montaggio precedente, e lo divideremo in due parti uguali.

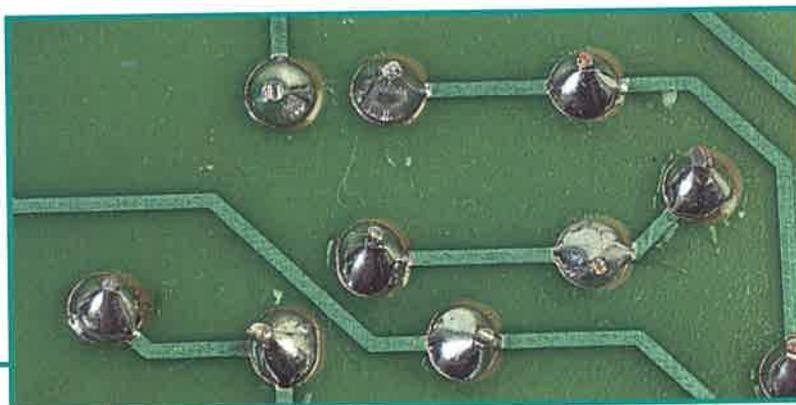
Dopo aver saldato i due fili sul microfono procederemo alla sua inserzione sulla scheda dei sensori. Lo inseriremo in X1 come mostrato nell'immagine.



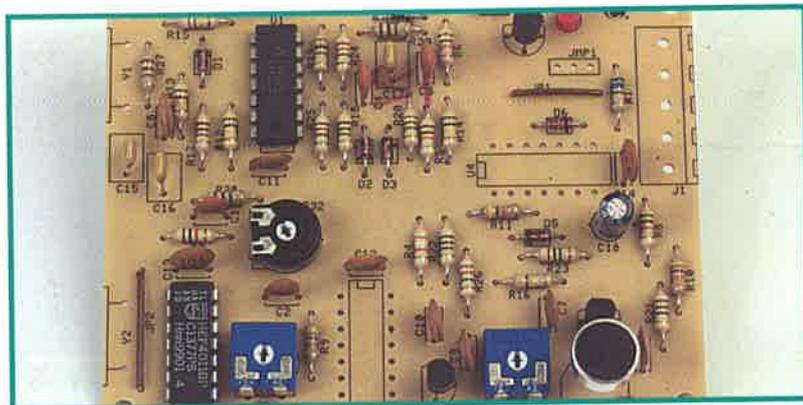


Continueremo saldando i due fili sulla scheda dei sensori. Ora disponiamo di un microfono montato e pronto per rilevare le soglie dei suoni.

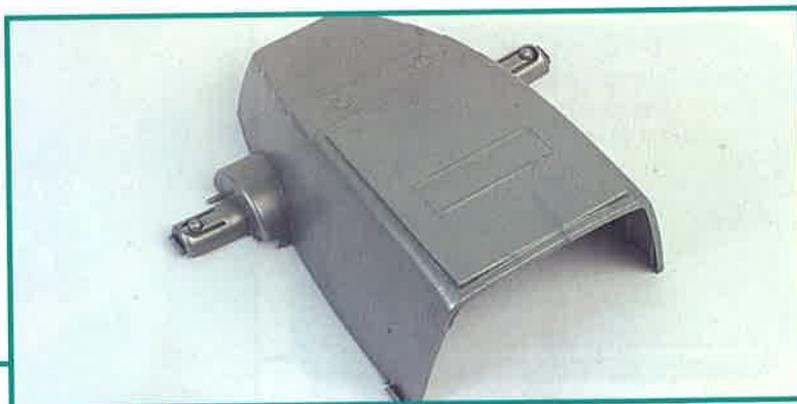
Questa è la forma che devono avere le saldature dei fili del microfono. Non dobbiamo riscaldare eccessivamente il filo rigido durante la saldatura, per non rischiare di fondere l'isolante.

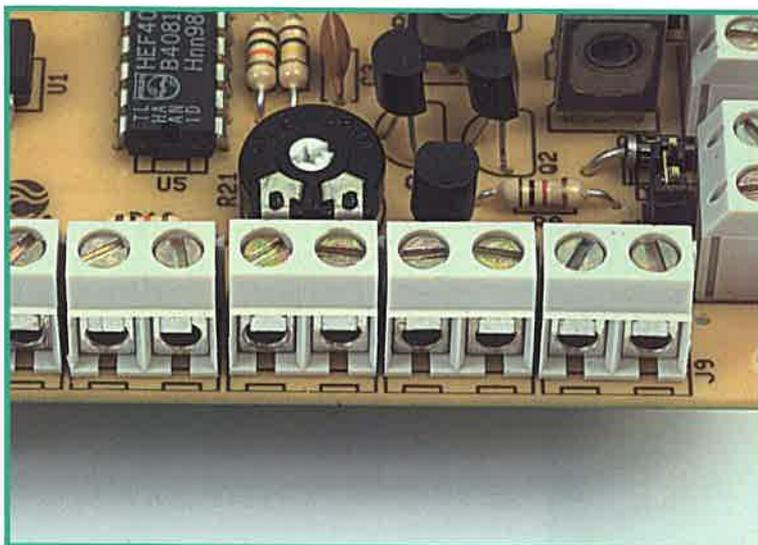


Questo è l'aspetto che presenta la scheda dei sensori con il microfono già montato. A questo punto conviene verificare la corretta inserzione di tutti i componenti montati sino a questo momento.



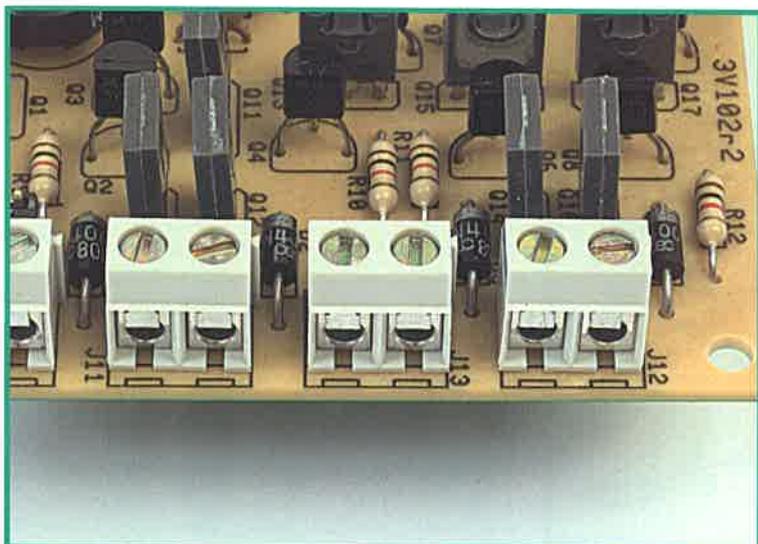
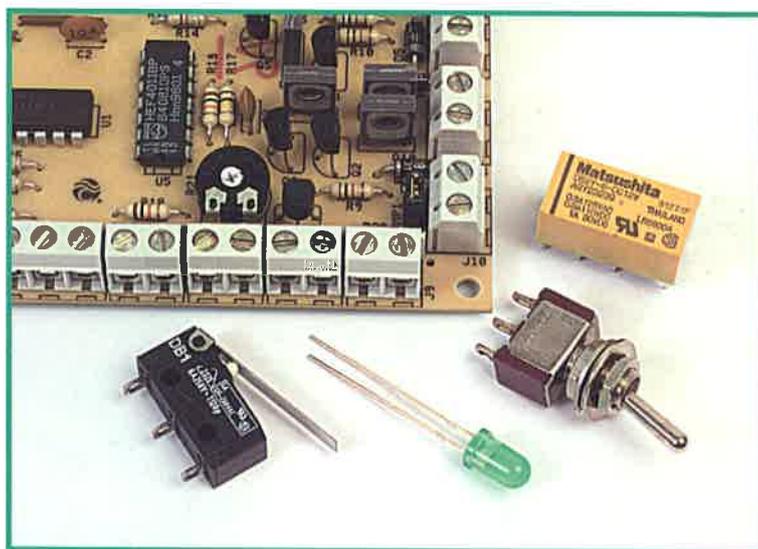
Questo è un pezzo del corpo del robot Monty. Lo dobbiamo conservare per poterlo assemblare con gli altri pezzi del corpo e completare la struttura del robot.



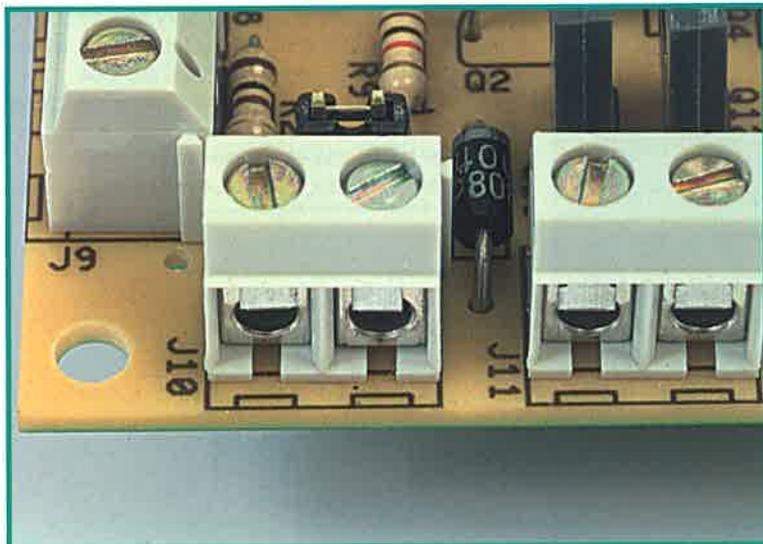


Oltre alle morsettiere di ingresso per i sensori ottici e meccanici disponiamo di altre quattro morsettiere a due terminali che ci serviranno come ingresso o uscita digitale. Sono J6, J7, J8 e J9; sono collegate ai piedini RB4, RB5, RB6 e RB7 del PIC, nell'ordine mostrato nell'immagine. Se guardiamo di fronte le morsettiere, il terminale di sinistra è la massa.

Possiamo utilizzare questa morsettiere per collegare qualsiasi tipo di dispositivo che vogliamo. Ad esempio, potremo montare diodi LED, o relè per attivare altri motori. Come ingressi potremo collegare più sensori meccanici come finecorsa, interruttori o pulsanti. Queste morsettiere inoltre servono per fare i collegamenti con i sensori della scheda dei sensori che fa parte di Monty.

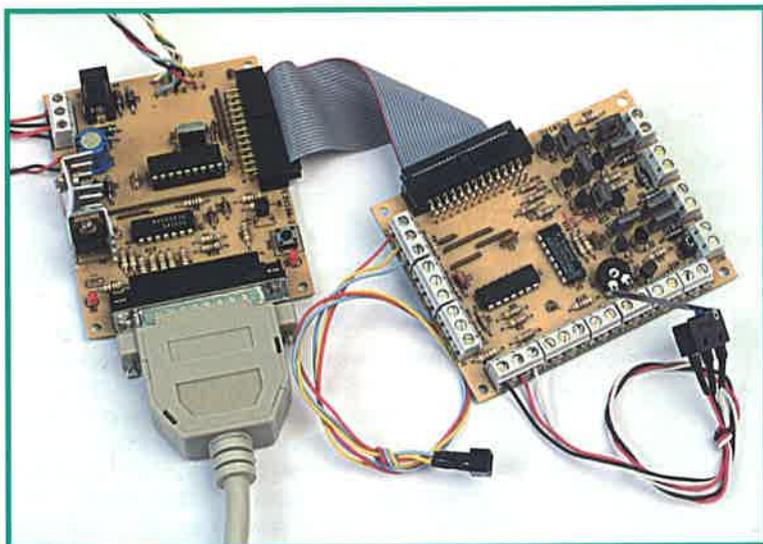
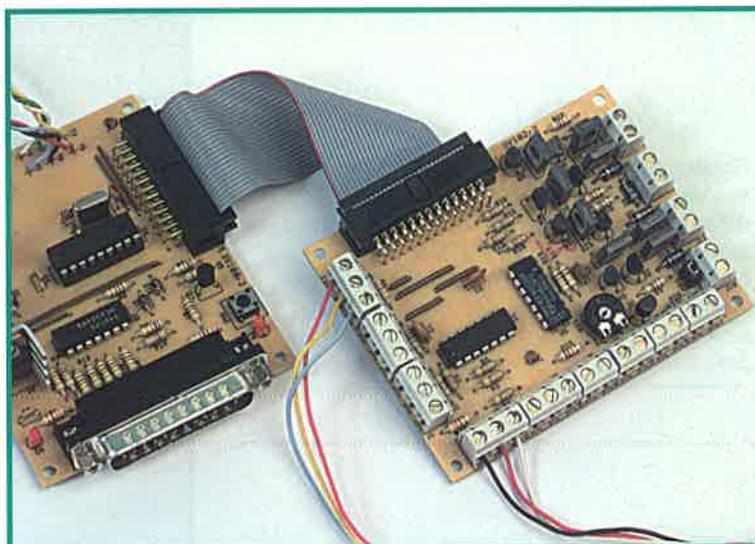


Le morsettiere J11, J12 e J13 serviranno per il controllo dei motori. Il motore che chiameremo M1 sarà collegato in J11 e il motore M2 in J12. La morsettiere J13, che si trova fra le due precedenti, servirà come ingresso della tensione di alimentazione per questi motori, dato che la scheda di potenza permette di disporre di un'alimentazione per i motori, indipendente dall'alimentazione della scheda. Guardando la morsettiere di fronte, il terminale di destra è la massa e in quello di sinistra verrà collegata la tensione positiva da applicare al motore.



Infine la morsettiere J10 serve per collegare un altoparlante, nello stesso modo in cui lo abbiamo fatto per la scheda di ingressi e uscite. Dobbiamo collegare il cavo rosso dell'altoparlante al segnale +5 V della morsettiere, che si trova nel terminale di sinistra, se guardiamo la morsettiere di fronte come mostra l'immagine.

Il collegamento fra la scheda di controllo, in cui risiede il PIC, e la scheda di potenza, lo realizzeremo attraverso il Pic-Bus, nello stesso modo con cui collegavamo la scheda di controllo con quella di ingressi e uscite. A partire da questo momento, utilizzeremo la scheda di potenza per valutare i programmi che funzioneranno con gli ingressi e le uscite reali con cui faremo funzionare il robot.

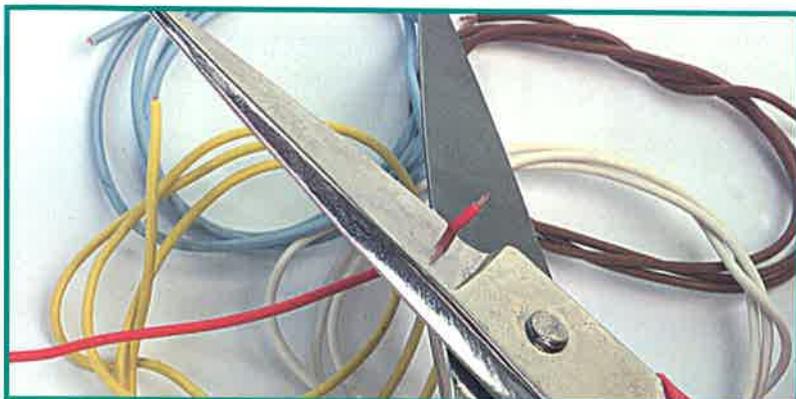


Per la scrittura del microcontrollore, non sarà necessario scollegare il Pic-Bus di connessione fra le due schede, sempre che non ci siano sensori collegati ai morsetti J8 e J9, dato che i segnali presenti su questi morsetti vanno a RB6 e RB7 cioè i piedini di scrittura del PIC. Come sempre faremo attenzione alla posizione dell'interruttore di scrittura della scheda di controllo, che deve essere in posizione PROG. Nell'immagine vediamo un insieme di sensori collegati alla scheda di potenza, e questa a sua volta collegata alla scheda di controllo.

MODULO DEI SENSORI

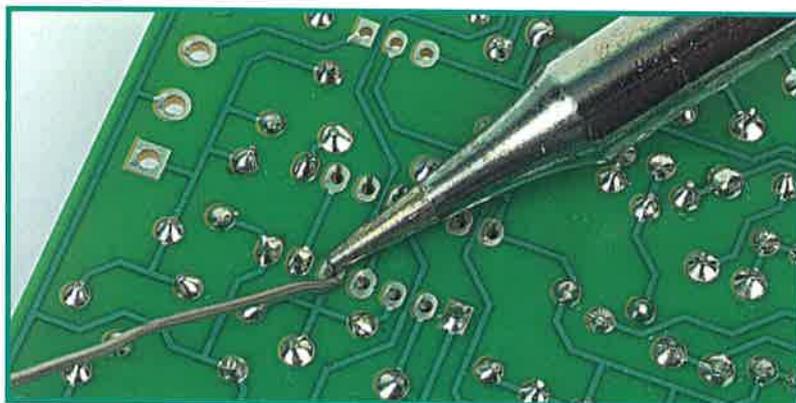
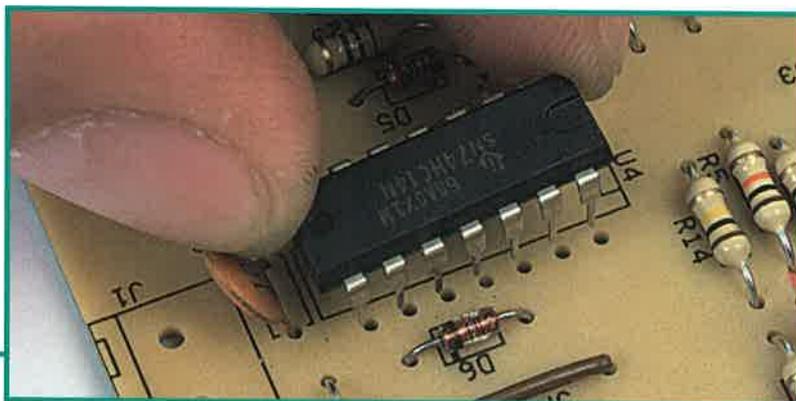
PL 80

Montaggio passo a passo



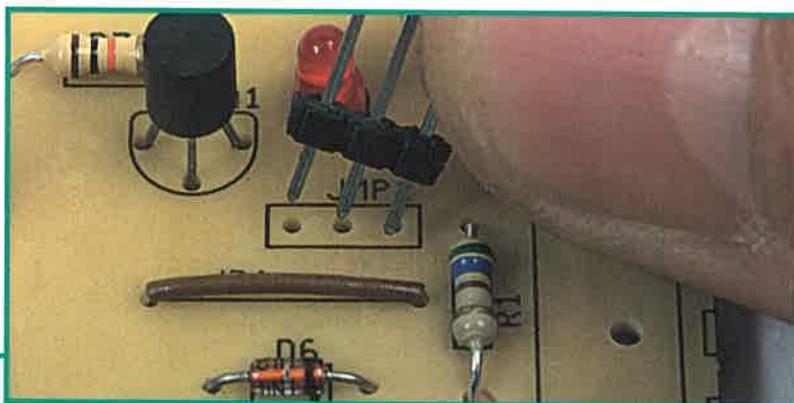
Questo insieme di cinque cavi colorati serve per collegare la scheda dei sensori con la scheda di potenza, tramite le morsettiere. Peleremo ognuno dei cavi da entrambi gli estremi.

Saldiamo ora il circuito integrato 74HC14 in U4. Dobbiamo far combaciare la tacca di riferimento dell'integrato con quella disegnata sulla serigrafia della scheda. È fondamentale orientarlo correttamente.

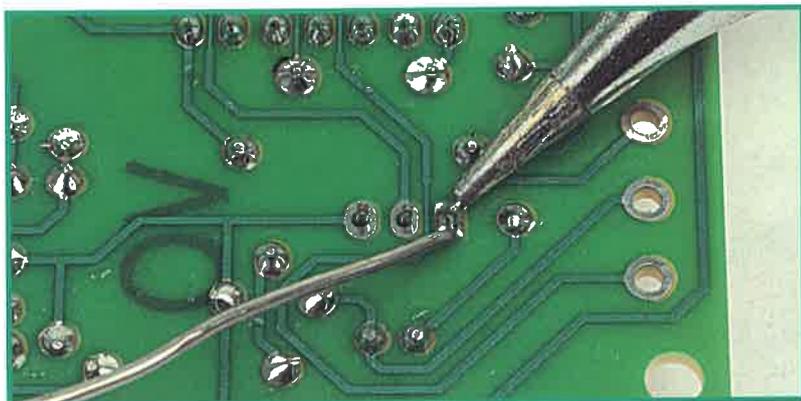


Dopo aver montato l'integrato salderemo i due pin agli angoli opposti; prima di continuare verificheremo che l'integrato sia ben inserito e orientato correttamente, poi salderemo il resto dei piedini.

Ora monteremo il connettore dritto a 3 vie nella posizione JMP1. La parte corta del connettore si inserirà nella scheda, lasciando la parte lunga rivolta verso l'alto.

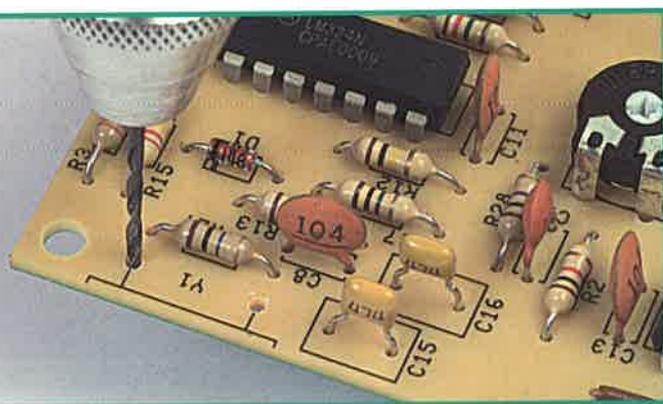


Montaggio passo a passo



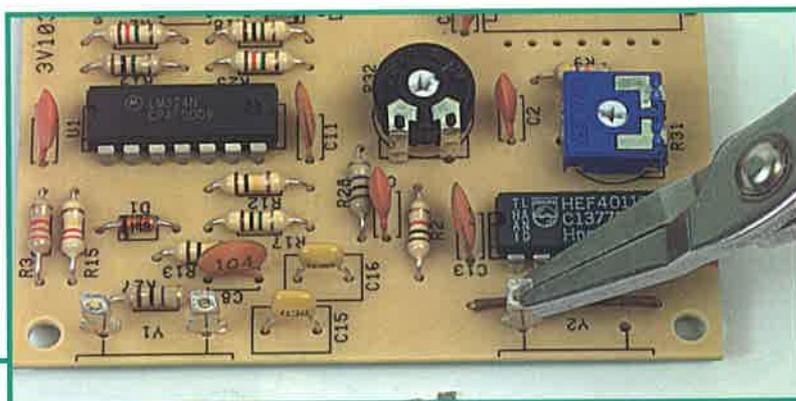
Si raccomanda di saldare solo un piedino, per poi verificare che il connettore non rimanga inclinato ma sia dritto e perfettamente appoggiato alla superficie. Poi salderemo il resto dei piedini.

Dopo aver saldato il connettore a 3 vie maschio, monteremo su esso il jumper a due posizioni per fare in modo che resti agganciato e che non si perda. In seguito impareremo che funzione ha il jumper, in ognuna delle due posizioni di chiusura.



Anche se potremmo saldare i cavi direttamente, disponiamo di 4 fastom, come opzione, per il collegamento della capsula ad ultrasuoni. Due di essi andranno in Y1, però prima ripasseremo i fori con una punta da 1,2 mm di diametro. Li monteremo spingendo sino a che siano totalmente inseriti, dopo li salderemo.

Gli altri due fastom li monteremo in Y2. Il processo di montaggio è simile a quello dei fastom di Y1: faremo pressione per farli entrare nella scheda il più possibile, dopo procederemo alla saldatura.



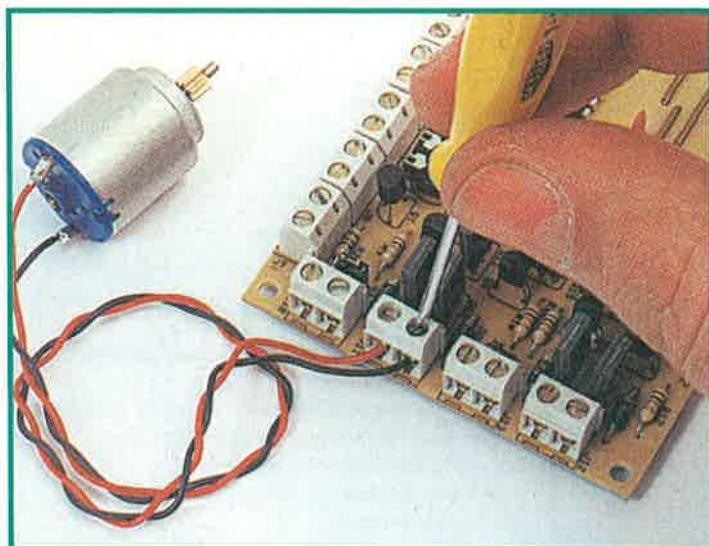
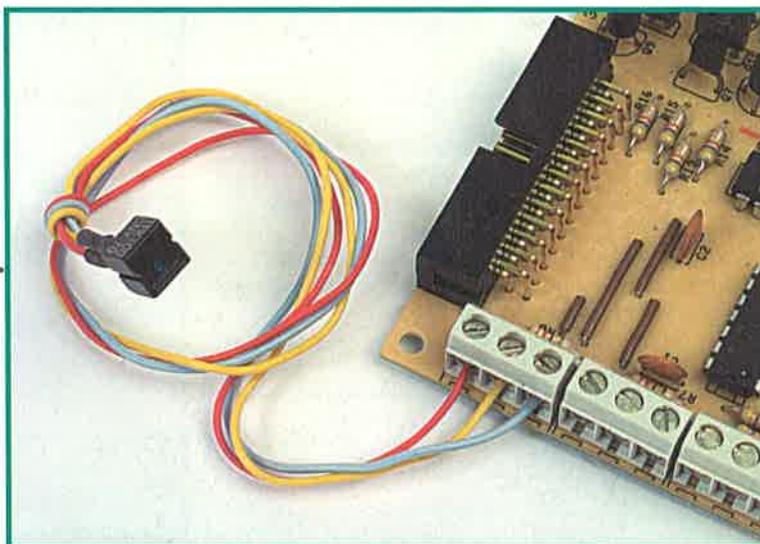
```
c:\progra~1\mplab\ase_41\esempio1.asm
1
2      List    p=16F84a
3      include "P16F84a.INC"
4
5      org    0x00
6      goto  Inizio
7      org    0x05
8
9 Inizio      bsf    STATUS,RP0
10     movlw b'00010000'
11     movwf  PORTA
12     bcf    STATUS,RP0
13
14 Ciclo     btfss  PORTA, 4
15     goto  Avanti
16     goto  Fermo
17
18 Fermo     c1rf   PORTA
19     goto  Ciclo
20
21 Avanti    bsf    PORTA,0
22     goto  Ciclo
23
24     END
```

Questo è il primo esercizio che faremo utilizzando sensori e motori che diventeranno parte integrante di Monty. L'esercizio consiste nell'accensione e nello spegnimento di un motore a corrente continua (che controllerà una delle due ruote di trazione del robot) secondo lo stato di un sensore ottico a riflessione. Sarà necessario utilizzare la scheda di controllo collegata alla scheda di potenza. Il programma è riportato nella figura e consiste nell'attivare / disattivare il motore 1 in funzione dello stato del sensore su RA4.

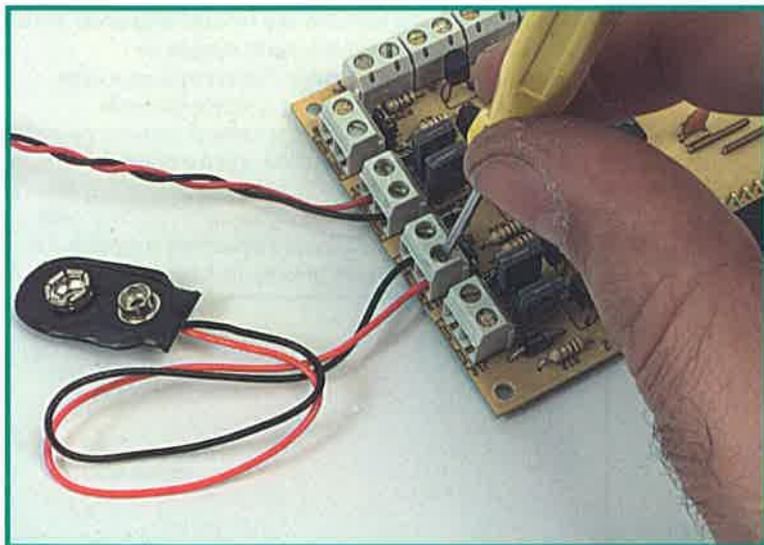
Realizzeremo il montaggio necessario sulla scheda di potenza per provare questo esercizio.

Collegheremo un sensore ottico sulla morsettiera J5, tramite il quale il segnale arriverà al pin RA4 del PIC. È necessario collegare correttamente i cavi nella morsettiera, come già sappiamo.

Se collegheremo male l'alimentazione del sensore potremmo bruciarlo e renderlo inutilizzabile.

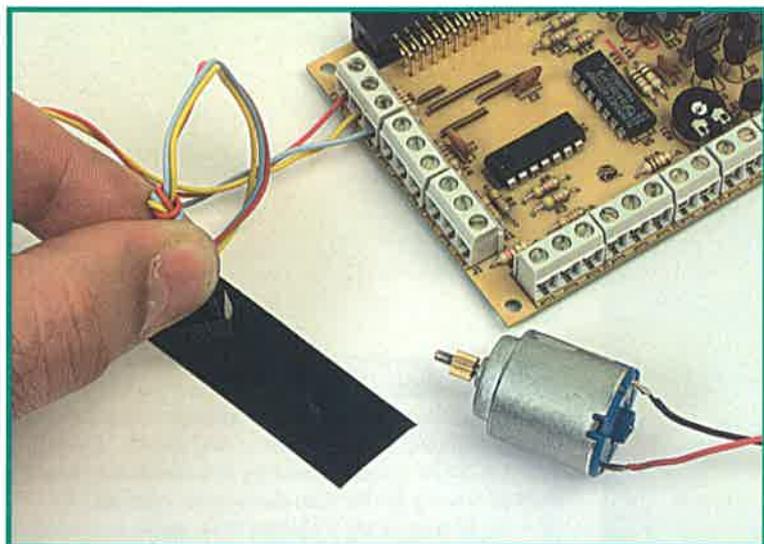
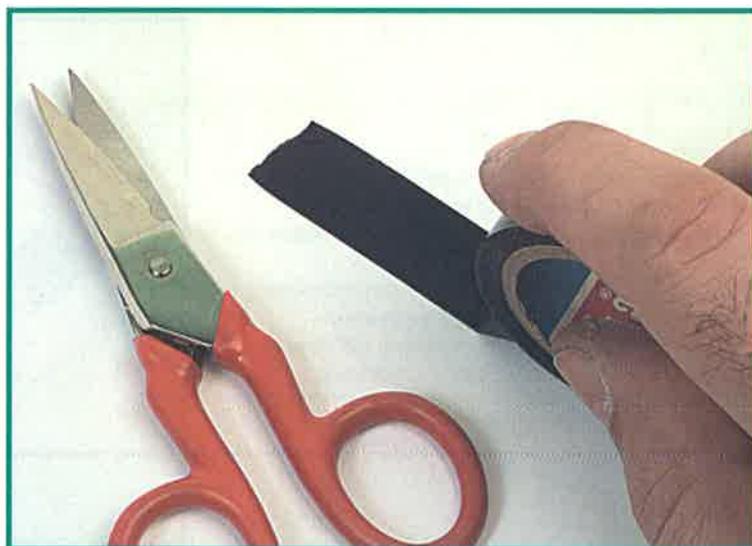


Oltre al sensore ottico collegheremo alla scheda di potenza un motore a corrente continua. Lo inseriremo nella morsettiera J11 (motore 1) tramite la quale il motore resterà collegato ai piedini RA0 e RA1 del PIC. Quando il valore di uscita di questi pin sarà due zeri o due uno, il motore rimarrà fermo. Con due segnali opposti "1 - 0" o "0 - 1", faremo girare il motore in un verso o in un altro.



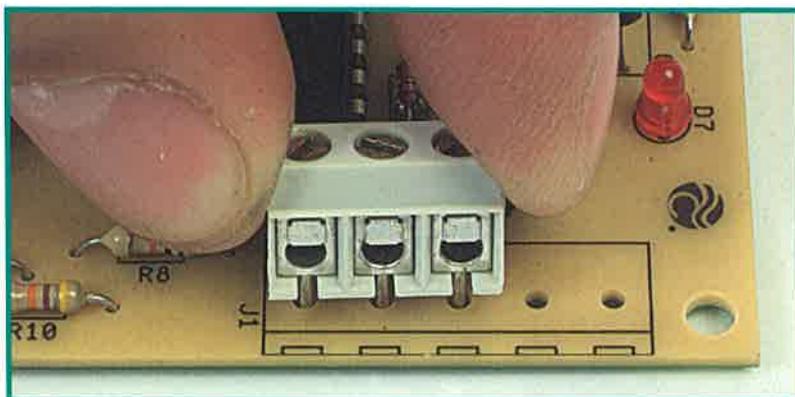
Oltre al collegamento del motore su J11, è necessario fornire tramite i morsetti J13 la tensione di alimentazione dei motori, che potrà essere erogata da una pila collegata alla clips. Dobbiamo fare in modo che la massa dell'elettronica sia comune alla massa che applicheremo ai motori, per cui dovremo inserire il terminale negativo del portatile (cavo nero) nella stessa posizione che vediamo in figura.

Per provare l'esercizio dobbiamo attivare e disattivare il sensore ottico. Potremo utilizzare un pezzo di nastro isolante nero su una superficie bianca (o riflettente). Quando il sensore si trova sul fondo scuro del nastro la luce emessa dal fotodiode non polarizzerà il fototransistor; quando si troverà sul fondo bianco la luce del fotodiode arriverà sino alla base del fototransistor e questo condurrà.



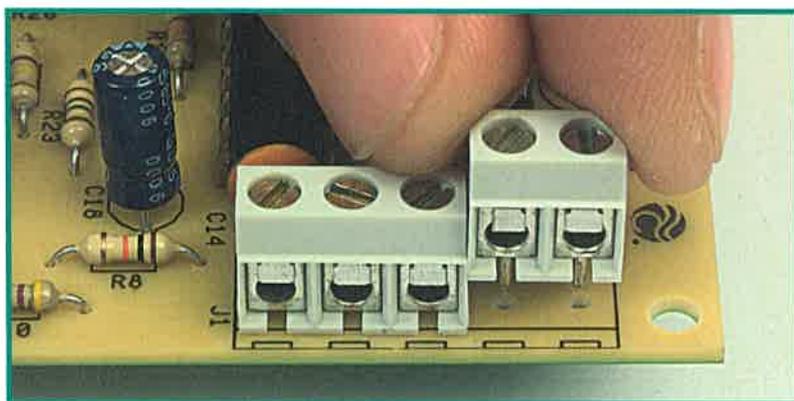
Scriveremo, compileremo e trasferiremo il programma sul microcontroller tramite la scheda di controllo. In seguito collegheremo la scheda di controllo con quella di potenza tramite il PicBus e proveremo l'esercizio. Raccomandiamo di ripetere l'esercizio, però con il sensore ottico collegato su morsetti diversi, e anche con i sensori meccanici, al fine di verificare il corretto montaggio della scheda di potenza.

Montaggio passo a passo



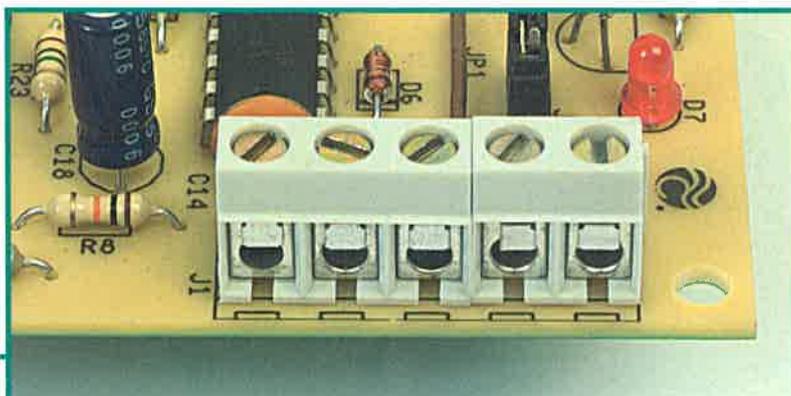
Monteremo una morsettieria a tre contatti nella parte sinistra del connettore J1. È necessario inserire la morsettieria completamente, sino a filo della scheda dei sensori.

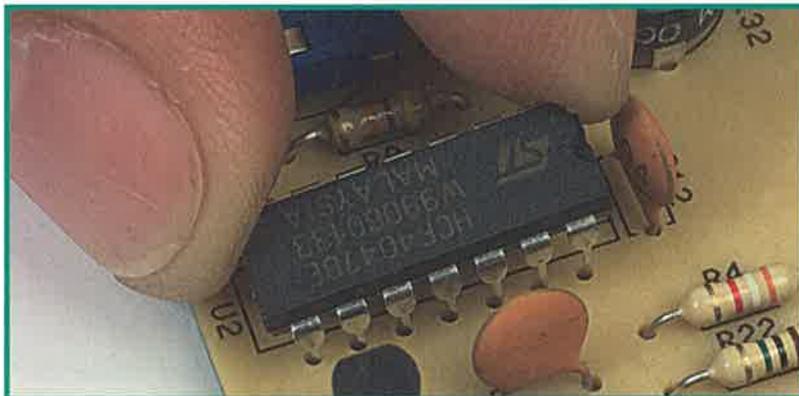
Per saldare le morsettiere dobbiamo utilizzare più stagno di quanto non facciamo normalmente per il resto dei componenti. Fonderemo lo stagno sufficiente per formare un cono appoggiato sulla superficie scheda, come mostrato in figura.



La morsettieria a due contatti va inserita a lato di quella da tre, come mostrato nell'immagine. L'uscita dei morsetti deve essere orientata verso l'esterno della scheda.

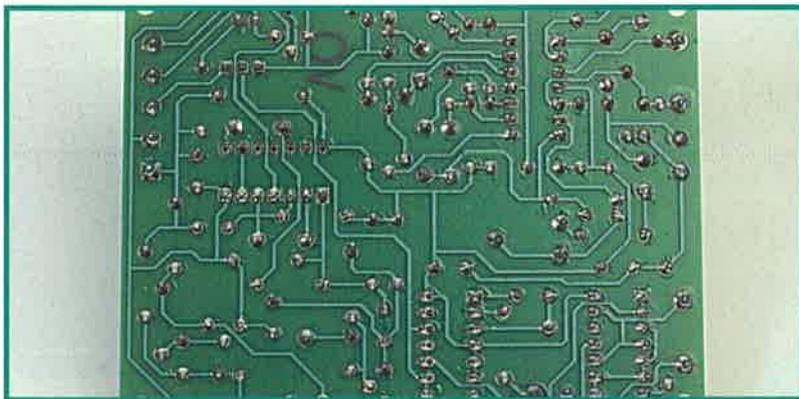
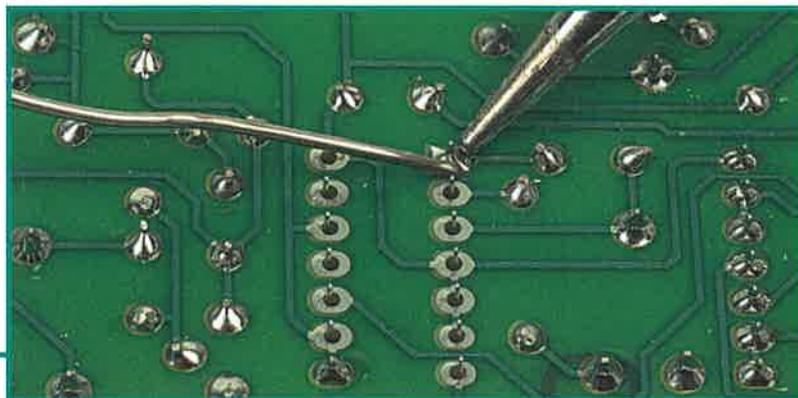
Prima di procedere alla saldatura della morsettieria dobbiamo assicurarci di averla inserita completamente e che sia a filo della scheda. Poi procederemo a saldare i suoi due piedini.





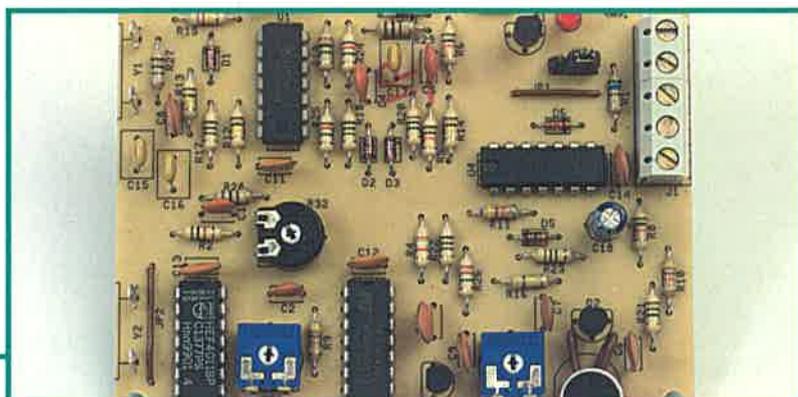
Ora monteremo il circuito integrato 4047 in U2. Dobbiamo far coincidere la tacca di riferimento del circuito integrato con quella disegnata sulla serigrafia della scheda. La tacca, come sempre, ci indica il verso di inserzione del chip.

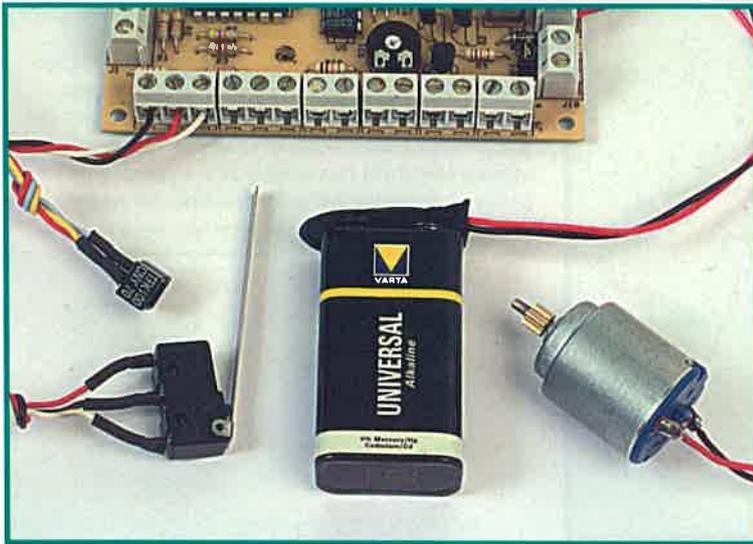
Inizialmente dobbiamo saldare solo due piedini del circuito, preferibilmente due angoli opposti. A quel punto prima di proseguire a saldare verificheremo di non aver sbagliato il verso di inserzione del chip, che deve essere totalmente a filo della scheda.



Dopo esserci assicurati del corretto montaggio del chip, salderemo il resto dei pin. Questo è l'aspetto che deve avere la scheda dei sensori con tutti i suoi componenti già inseriti e saldati.

Nella figura vediamo la scheda dei sensori con tutti i suoi componenti. Verificare che siano inseriti nelle loro rispettive posizioni e che i versi dei componenti con polarità siano appropriati.





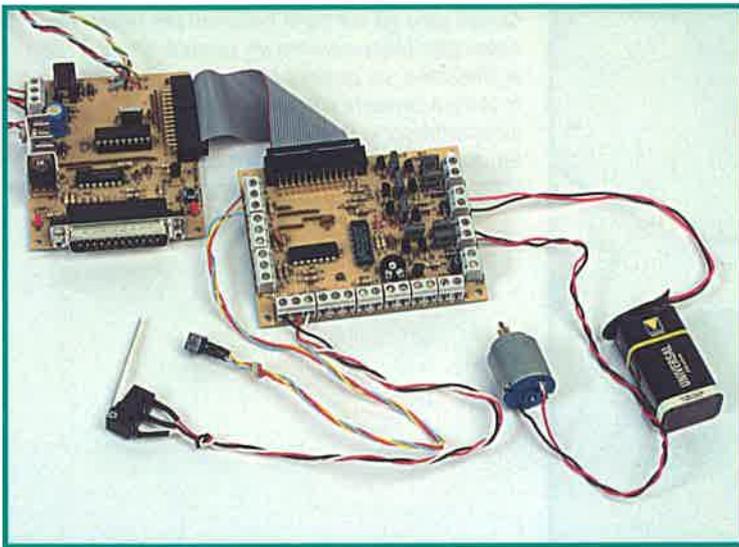
Questi sono gli elementi necessari per realizzare l'esercizio: combineremo un sensore ottico a riflessione, un sensore meccanico (finecorsa) e un motore a corrente continua. Inoltre avremo bisogno di una clips per pile per fornire la tensione di alimentazione al motore.

In questo esercizio simuleremo l'avanzamento del robot che segue un percorso, controllando nel contempo che non vada a urtare. Per seguire il percorso o pista, utilizzeremo un pezzo di nastro isolante nero su una superficie riflettente, e un sensore ottico a riflessione. Per rilevare le collisioni utilizzeremo un sensore meccanico tipo finecorsa. Nella figura è riportata l'intestazione del programma, dove configuriamo le linee di ingresso e di uscita.

```
c:\progra~1\mplab\fas_42\eserciz1.asm
1
2
3 List p=16F84
4 include "PIC16F84A.INC"
5
6 org 0x00
7 goto Inizio
8 org 0x05
9 Inizio bsf STATUS, RP0
10 movlw b'00010000'
11 movwf PORTA
12 movlw b'00000100'
13 movwf PORTB
14 bcf STATUS, RP0
15
```

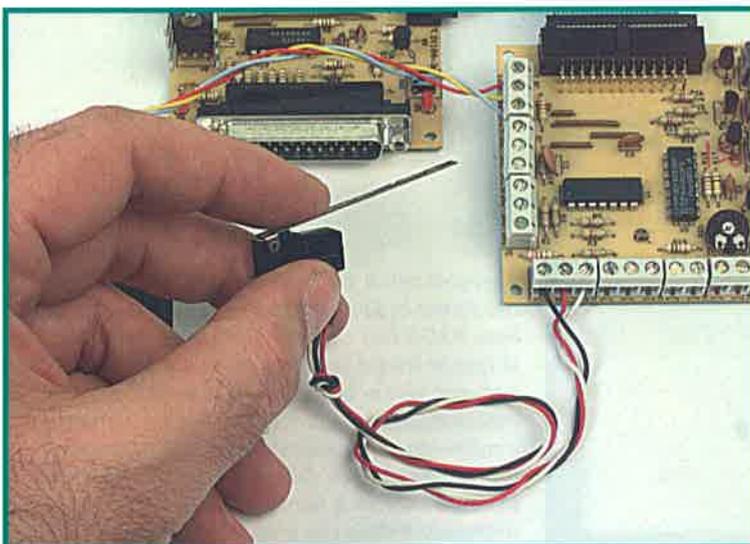
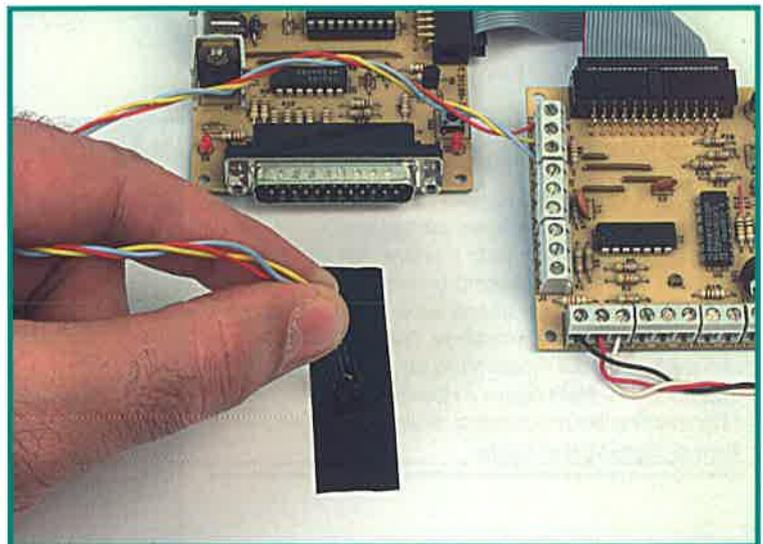
```
c:\progra~1\mplab\fas_42\eserciz1.asm
16
17 Ciclo btfsc PORTB, 2
18 goto FERMA
19 btfss PORTA, 4
20 goto FERMA
21 goto AVANTI
22
23 FERMA clrf PORTA
24 goto Ciclo
25
26 AVANTI bsf PORTA, 0
27 goto Ciclo
28
29 END
```

Il sensore ottico si collegherà in RA4, e il sensore meccanico in RB2. Il motore si controllerà tramite le linee RA0 e RA1 (Motore 1). Quando il valore di queste linee è uguale, il motore si ferma, e quando è diverso gira in un determinato verso. Il sensore meccanico ha priorità sul programma, fermando il movimento del motore se passa allo stato attivo, indipendentemente dallo stato del sensore ottico. Inoltre, il motore si ferma anche se il sensore a riflessione rileva che è stata abbandonata la pista.



Collegiamo il sensore ottico nella morsettiera J5 per fare in modo che arrivi al pin RA4. Il sensore meccanico si collegherà in J3, tramite il quale il suo segnale arriverà su RB2. Il motore sarà collegato alla morsettiera J11, in modo che sia controllato dai segnali RA0 e RA1. Forniremo alimentazione al motore tramite la morsettiera J13. È necessario collegare la massa dell'alimentazione del motore in comune con la massa dell'elettronica della scheda.

Per simulare il robot che segue il tracciato possiamo utilizzare un pezzo di nastro isolante nero messo sopra una superficie bianca. Sopra il nastro collocheremo il sensore ottico CNY70. Quando il sensore è sopra il nastro, il motore si attiva simulando l'avanzamento del robot, se il sensore esce dalla linea nera il motore si ferma.

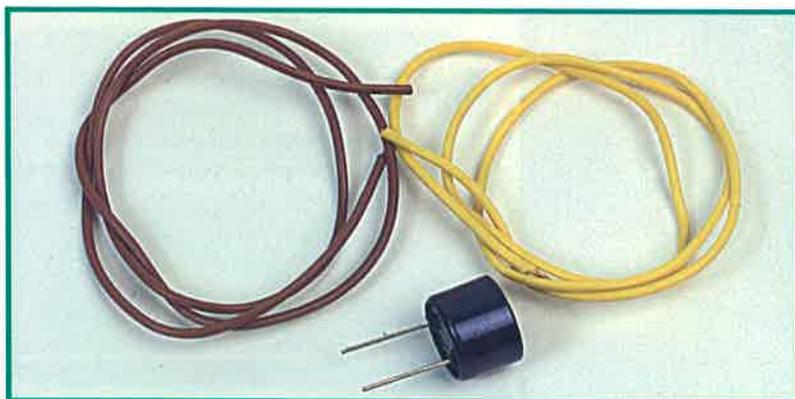


Nel caso che il sensore meccanico si attivi, il motore si fermerà, sia il sensore ottico sopra la pista oppure no. In questo modo simuliamo la fermata del microrobot in caso di collisione contro qualche ostacolo, mentre sta percorrendo il tracciato. Con questo metodo utilizzeremo i finecorsa per rilevare gli oggetti o per evitare dei colpi al robot, frenandolo prima che si verifichi lo scontro.

MODULO DEI SENSORI

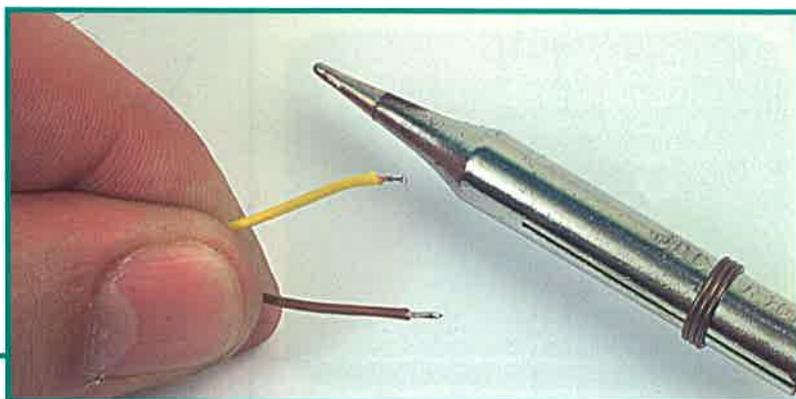
PL 84

Montaggio passo a passo



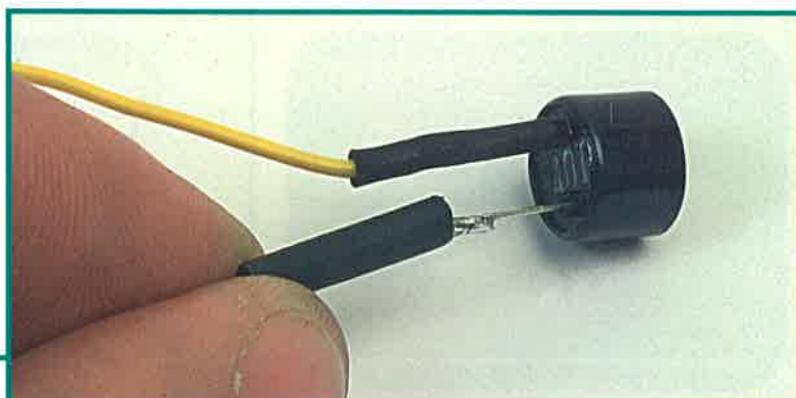
Saldiamo ora una delle due capsule ad ultrasuoni della scheda dei sensori, la capsula trasmittente. Avremo bisogno dei cavi mostrati in figura, che sono stati forniti nei fascicoli precedenti.

In primo luogo dobbiamo preparare i cavi. Li pareggeremo in lunghezza e li speleremo ai due estremi. Inoltre li prestagneremo per facilitare le saldature.

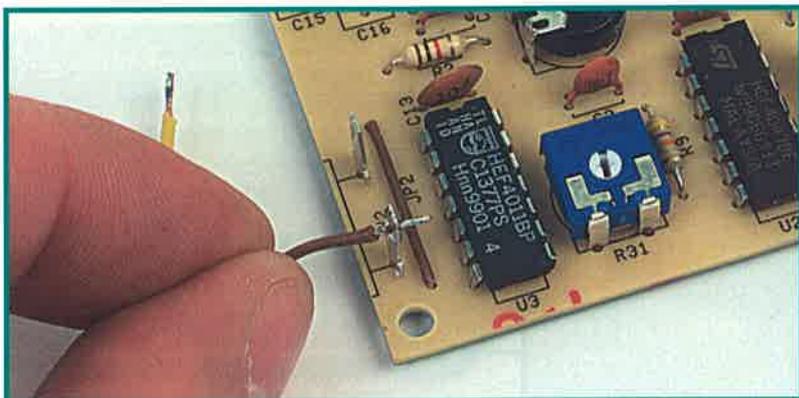


Prestagneremo entrambi i pin della capsula trasmittente ad ultrasuoni; dobbiamo lasciare una piccola goccia di stagno su entrambi i pin.

Ora uniremo i cavetti già prestagnati con i pin, prestagnati anch'essi, della capsula ad ultrasuoni. Per fare questo uniremo i cavetti ai pin, e riscaldaremo la zona con il saldatore. Infine monteremo due pezzi di tubetto termorestringente, precedentemente tagliati, ad una lunghezza di circa 1,5 cm.

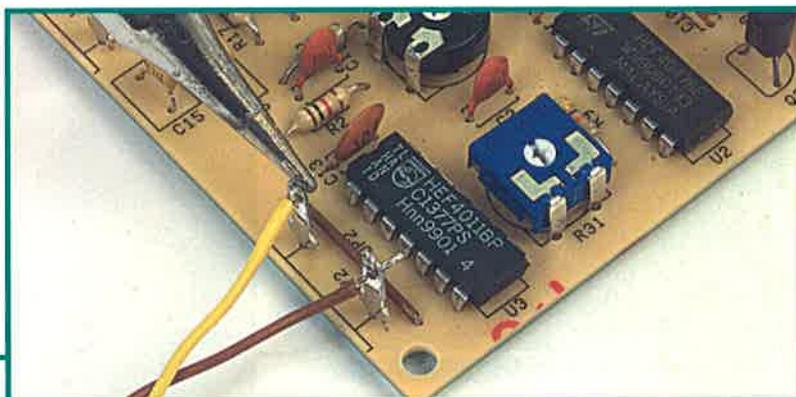


Montaggio passo a passo

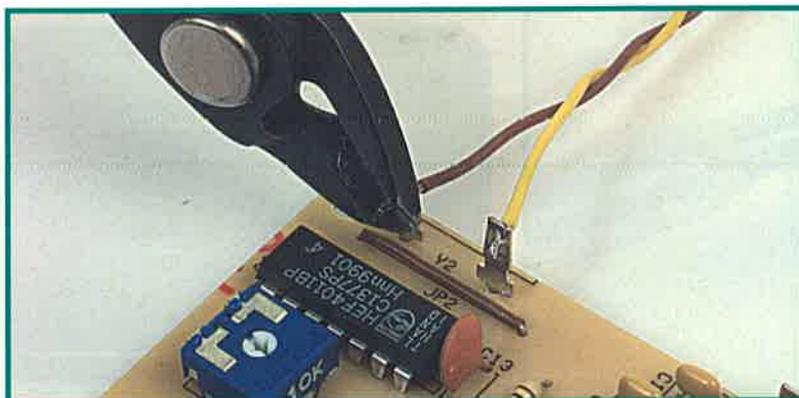


Ora dobbiamo saldare gli altri due estremi dei cavetti alla scheda dei sensori, direttamente o utilizzando i fastom che si trovano in Y2. La capsula ad ultrasuoni non ha polarità, quindi è indifferente l'ordine di collegamento dei cavetti.

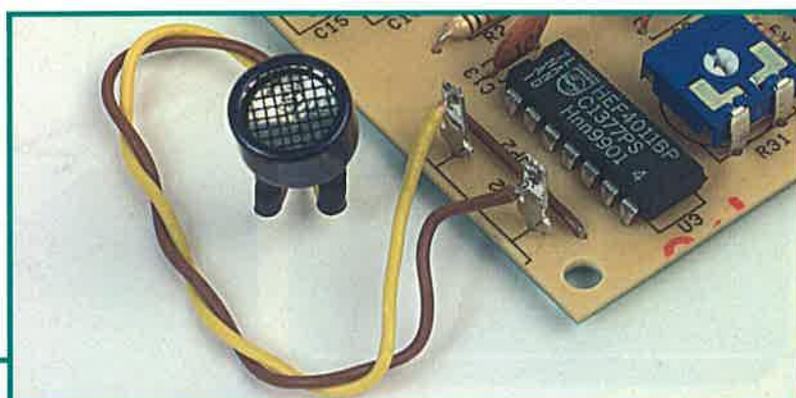
Per saldare i cavetti li introdurremo, già prestagnati, nei fori dei fastom, poi riscaldaremo la zona con il saldatore, e il cavo si salderà al fastom.

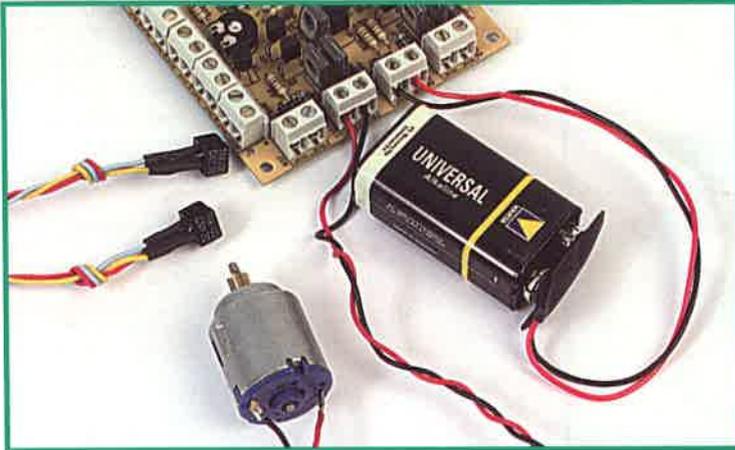


Se sarà necessario, taglieremo con un tronchesino gli estremi dei cavetti che fuoriescono dai fastom.



Questo è l'aspetto che presenta il trasmettitore di ultrasuoni, preparato e saldato alla scheda dei sensori. Quando salderemo la capsula ricevente avremo completato il montaggio della scheda dei sensori.





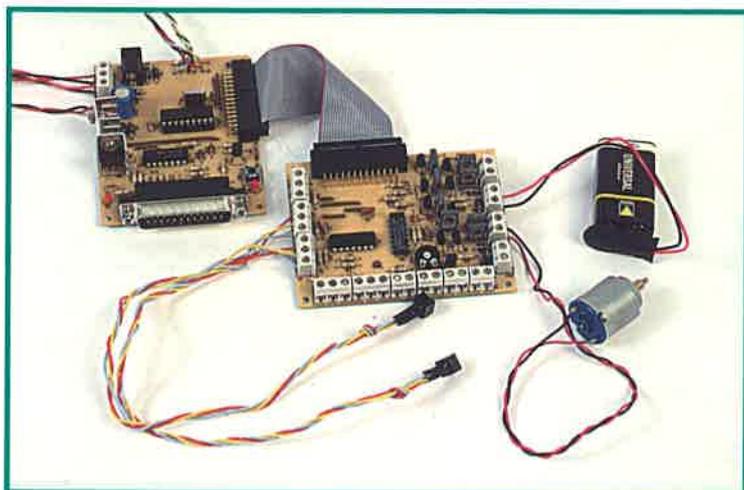
Realizzeremo un programma con cui controlleremo il verso di rotazione del motore, facendo in modo che possa girare in un senso o nell'altro in funzione dello stato dei due sensori ottici. Questa sarà la base del movimento del robot: faremo in modo che i motori ruotino in senso opposto, quando vorremo far girare Monty di lato. Spesso le svolte saranno eseguite in funzione dello stato dei sensori ottici.

Per prima cosa dobbiamo selezionare il dispositivo, realizzare le definizioni e posizionare i vettori come già sappiamo. Configureremo le linee RB0 e RB1 come ingresso per i sensori ottici, faremo una serie di maschere per conoscere lo stato dei sensori e decidere se far muovere o no il motore, e con quale senso di rotazione.

```
c:\progra~1\mplab\asc_43\eserci1.asm
1
2      List    p=16F84a
3      include "P16F84a.INC"
4
5      org    0x00
6      goto  Inizio
7      org    0x05
8
9 Inizio:      bsf    STATUS,RP0
10           clr   PORTA
11           movlw b'00000001'
12           movwf PORTB
13           bcf   STATUS,RP0
14
15 CICLO:      btfs  PORTB,0
16           goto  NERO_DX
17           goto  BIANCO_DX
18
```

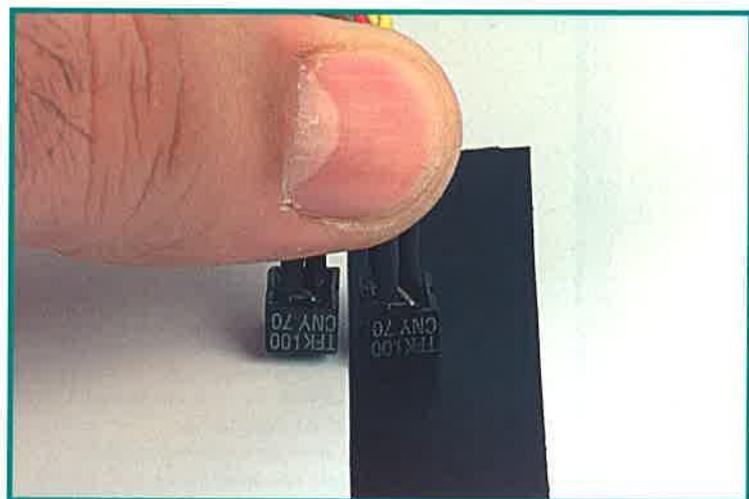
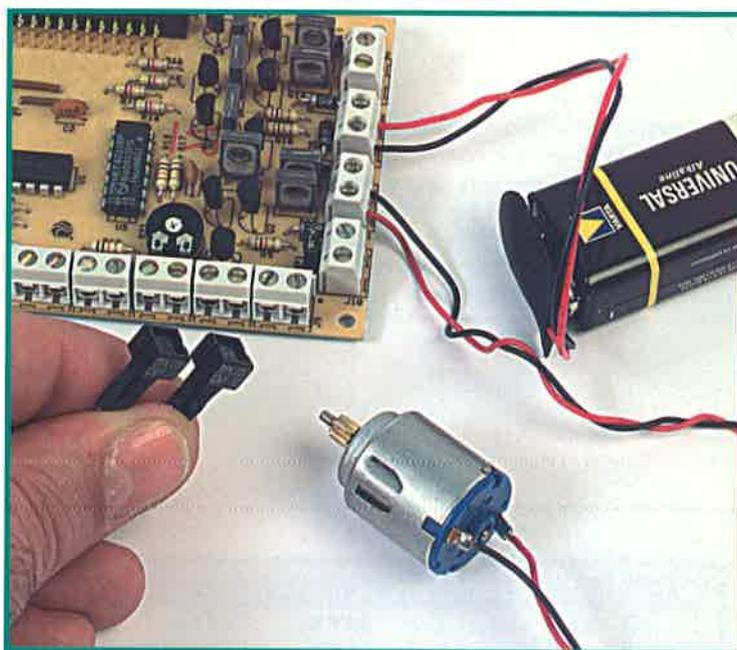
```
c:\progra~1\mplab\asc_43\eserci1.asm
19 NERO_DX:   btfs  PORTB,1
20           goto  FERMARE
21           goto  DESTRA
22
23 BIANCO_DX: btfs  PORTB,1
24           goto  SINISTRA
25           goto  FERMARE
26
27 FERMARE:   clr   PORTA
28           goto  CICLO
29
30 DESTRA     bsf   PORTA,0
31           bcf   PORTA,1
32           goto  CICLO
33
34 SINISTRA  bcf   PORTA,0
35           bsf   PORTA,1
36           goto  CICLO
37
38           END
```

Se entrambi i sensori sono inattivi o attivi, il motore rimarrà fermo; per fare in modo che si muova è necessario che un sensore sia attivo e l'altro no. A seconda di qual è quello attivo, faremo girare il motore in un verso o nell'altro. Il motore sarà collegato alle linee RA0 e RA1 del PIC.



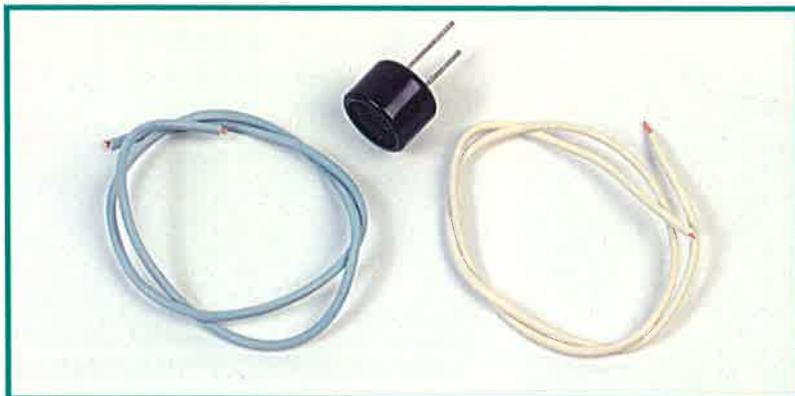
Collegeremo i sensori ottici nella morsettiere J1 e J2, in questo modo il segnale arriverà a RB0 e RB1. Il motore si collegherà alla morsettiere J11, per cui il controllo sarà realizzato tramite i segnali RA0 e RA1. È necessario fornire l'alimentazione dei motori tramite la morsettiere J13 della scheda di potenza, e collegarne la relativa massa in comune a quella dell'elettronica.

Se alziamo i due sensori, e li rivolgiamo verso l'alto, il motore si fermerà. Questo è dovuto al fatto che per i sensori il vuoto equivale ad una superficie totalmente nera. Potremo utilizzare questo trucco in seguito, al momento di mettere a punto i programmi, quando i sensori ottici si troveranno già fissati sulla struttura di Monty.



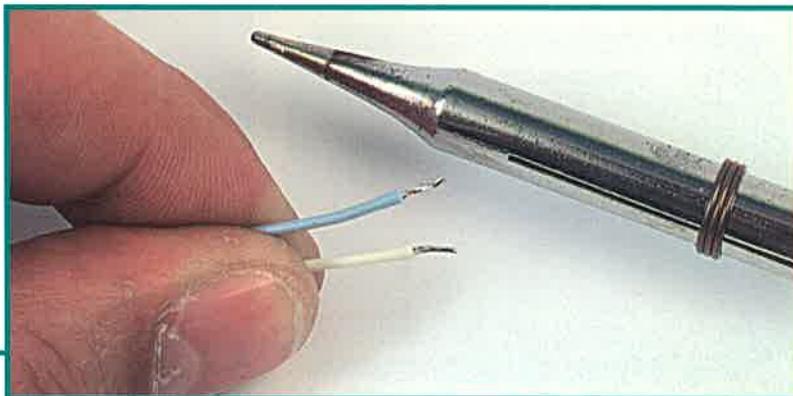
Per fare in modo che il motore giri in un verso oppure in un altro dobbiamo far sì che un sensore sia attivato e l'altro no. Questo si può ottenere unendo i due sensori e posizionandoli su una linea nera. Muoveremo i sensori verso sinistra e verso destra rispetto alla linea, in modo che uno si trovi dentro e l'altro fuori. A seconda di quale dei due rimarrà fuori dalla linea, il motore si girerà in un verso o nell'altro.

Montaggio passo a passo



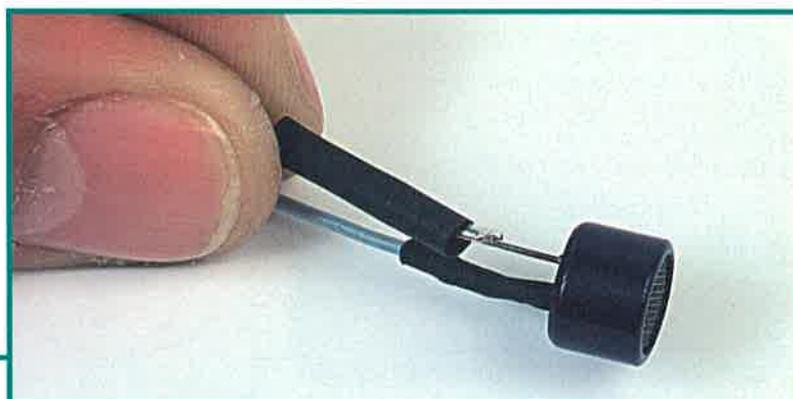
Saldiamo ora la capsula ricevente ad ultrasuoni che fa coppia con la capsula trasmittente che abbiamo già montato sulla scheda dei sensori. Avremo bisogno di due cavetti che sono stati forniti nei fascicoli precedenti.

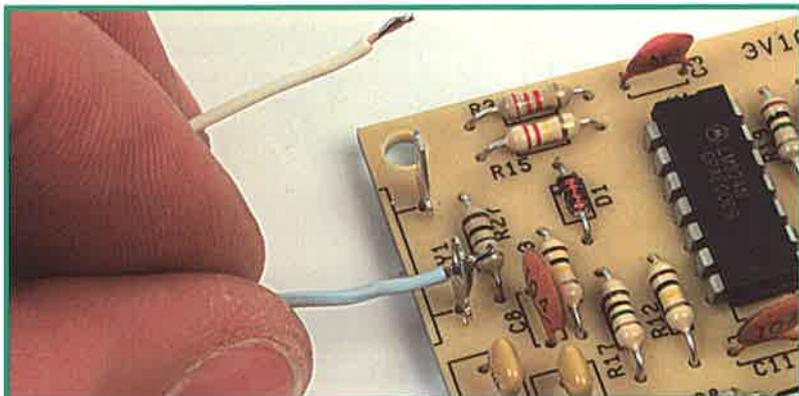
Peleremo i cavetti alle due estremità, e li prestagneremo come mostrato dall'immagine. Non è necessario utilizzare molto stagno, solamente la quantità sufficiente per far assumere al cavo un aspetto brillante.



Dovremo fondere lo stagno sui piedini della capsula ad ultrasuoni per facilitarne la saldatura. Per questo depositeremo un po' di stagno sul piedino della capsula e ne fonderemo una piccola goccia con il saldatore.

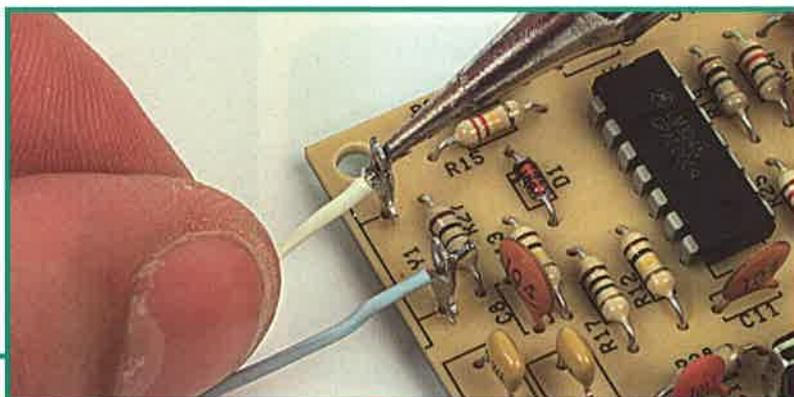
Ora salderemo i cavetti prestagnati alla capsula ad ultrasuoni. Non sarà necessario aggiungere altro stagno a quello che già si trova sui terminali della capsula ad ultrasuoni e sui cavetti.



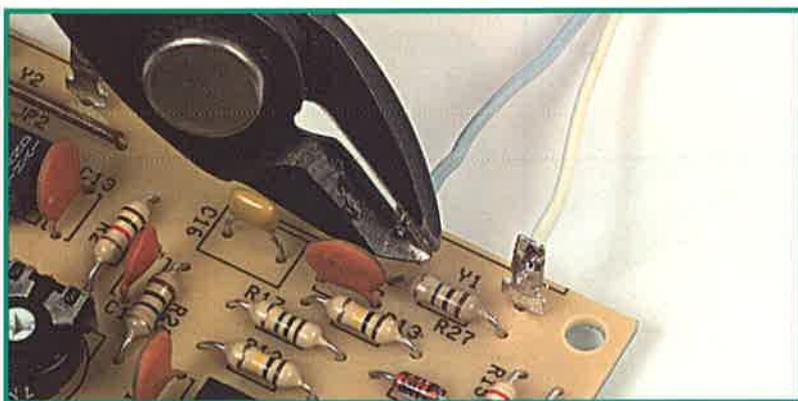


Dopo aver saldato uno degli estremi dei cavetti alla capsula, dobbiamo saldare l'altro alla scheda dei sensori; dovremo prestare anche questo estremo dei cavetti.

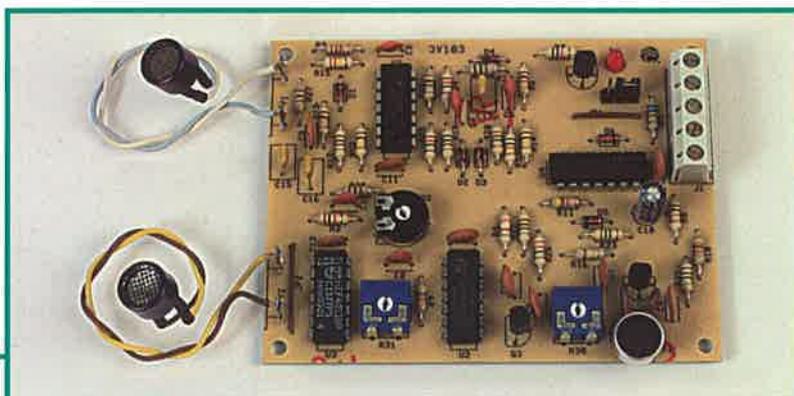
Per saldare i cavetti alla scheda dei sensori, utilizzeremo i fastom. Introduciamo il filo attraverso il foro del fastom e salderemo sino a che sia ben collegato. La capsula ricevente ad ultrasuoni andrà su Y1.

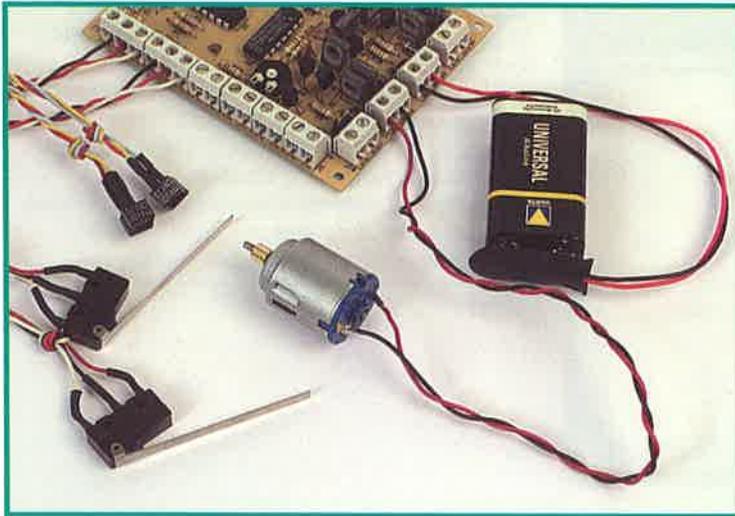


Nel caso fosse necessario, dovremo tagliare con un tronchesino l'eccesso di filo che fuoriesce dai fastom.



Questo è l'aspetto che presentano le due capsule ad ultrasuoni saldate alla scheda dei sensori, scheda che abbiamo ultimato. Ci servirà per fare in modo che Monty possa rilevare il movimento e il rumore acustico.





In questo esercizio combineremo i due sensori ottici a riflessione CNY70 con due sensori meccanici tipo finecorsa. Realizzeremo la simulazione del movimento del robot impiegando un motore e i sensori ottici per seguire una linea nera, in modo che il motore si attivi quando i sensori si trovano entrambi sulla linea nera, e si disattivi quando uno dei due esce fuori. Se i sensori meccanici si attivano (cioè se il robot entra in collisione con qualcosa), il robot girerà in senso contrario, simulando la retromarcia.

Per prima cosa definiremo il tipo di dispositivo e i registri interessati. Abbineremo i primi quattro pin della porta B (RB0-RB3), ai quattro sensori di ingresso che utilizzeremo. I sensori ottici andranno su RB0 e RB1 e quelli meccanici su RB2 e RB3. La porta A funzionerà come uscita, e il motore sarà collegato a RA0 e RA1.

```
c:\progra~1\mplab\fac_44\eserc1.asm
1
2
3
4
5
6
7
8
9 Inizio:
10
11
12
13
14
List p=16F84
include "P16F84a.INC"

org 0x00
goto Inizio
org 0x05

bsf STATUS,RP0
clrf PORTA
movlw b'00001111'
movwf PORTB
bcf STATUS,RP0
```

```
c:\progra~1\mplab\fac_44\eserc1.asm
15 ;*****
16
17 CICLO      btfsc  PORTB,2
18           goto   INDIETRO
19
20           btfsc  PORTB,3
21           goto   INDIETRO
22
23           btfss  PORTB,0
24           goto   FERMARE
25
26           btfss  PORTB,1
27           goto   FERMARE
28           goto   AVANTI
29
```

In seguito definiremo le maschere che realizzeremo con i segnali dei sensori, per decidere che tipo di movimento applicare. Prima verificheremo i sensori meccanici, dato che hanno priorità, e se si attivano dobbiamo fare in modo che il motore giri indietro. Poi verificheremo i sensori ottici, e a seconda che siano o meno sulla linea nera, decideremo se fermare o attivare il motore.

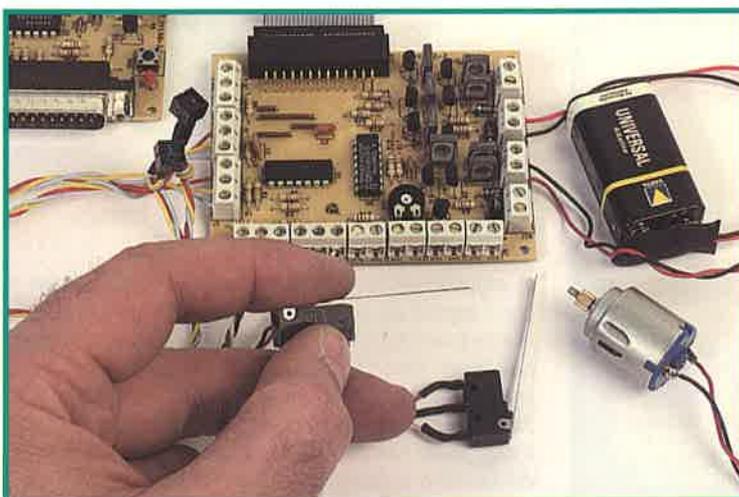
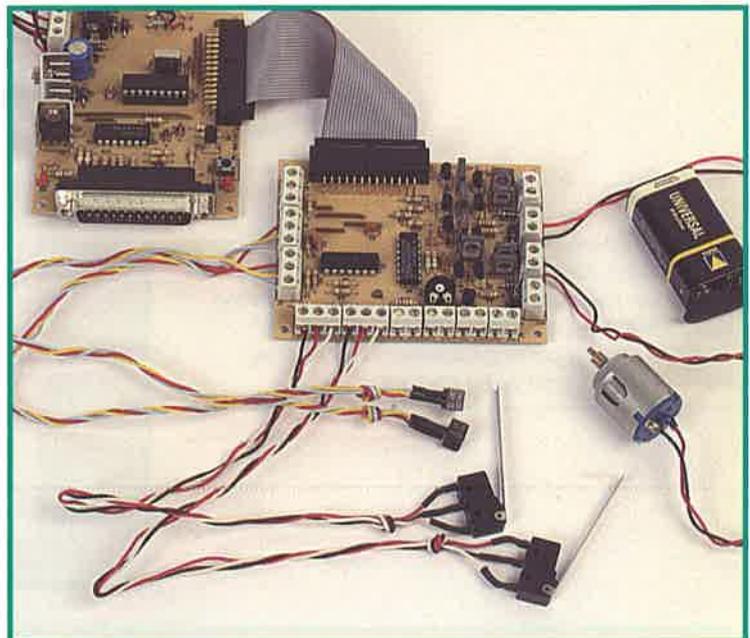
```

c:\progra~1\mplab\asc_44\eserci1.asm
30 ;*****
31
32 FERMARE      clr  PORTA
33             goto CICLO
34
35 AVANTI       bsf  PORTA,0
36             bcf  PORTA,1
37             goto CICLO
38
39 INDIETRO     bsf  PORTA,1
40             bcf  PORTA,0
41             goto CICLO
42
43             END

```

Queste routines sono quelle che servono per far muovere il motore. Quando i segnali in RA0 e RA1 sono uguali (due zeri o due uno) il motore si ferma. Con valori opposti, '1-0' o '0-1', otterremo di far muovere il motore in un verso o nell'altro; la rotazione del motore dipende anche da come lo collegheremo alle morsettiere della scheda di potenza. Scambiando i fili otterremo che il verso dell'avanzamento sia secondo un senso di rotazione o l'altro.

Questi sono i collegamenti che dovremo realizzare per provare l'esercizio dopo che il programma sia stato editato, compilato e scritto nella memoria del microcontroller. Collegheremo i due sensori ottici in J1 e J2, i due sensori meccanici in J3 e J4, e il motore sui morsetti di J11. È necessario fornire una tensione per alimentare il motore in J13 (come sempre, la massa dell'alimentazione del motore dovrà essere in comune con quella dell'elettronica).

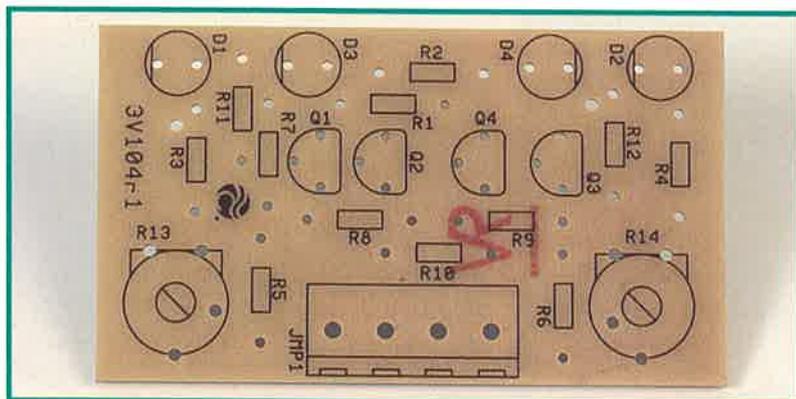


Per verificare l'esercizio collocheremo una linea nera su uno sfondo bianco o riflettente. In seguito controlleremo il funzionamento del programma con i sensori ottici. Esaminato questo punto passeremo a provare i sensori meccanici, per accertarci che il motore cambi senso di rotazione se vengono attivati.

MODULO DELLE LUCI

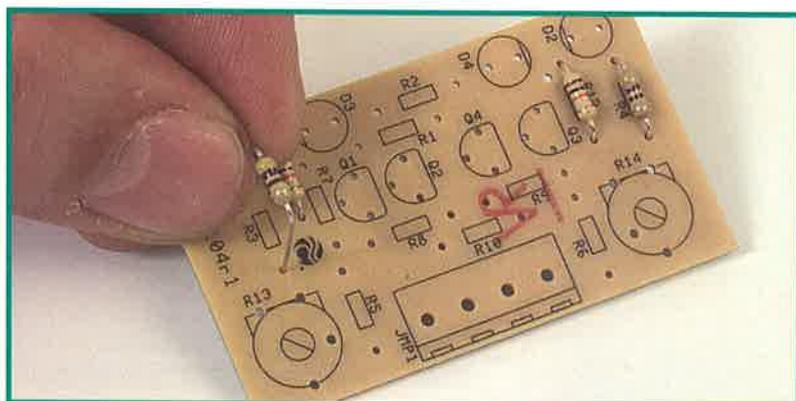
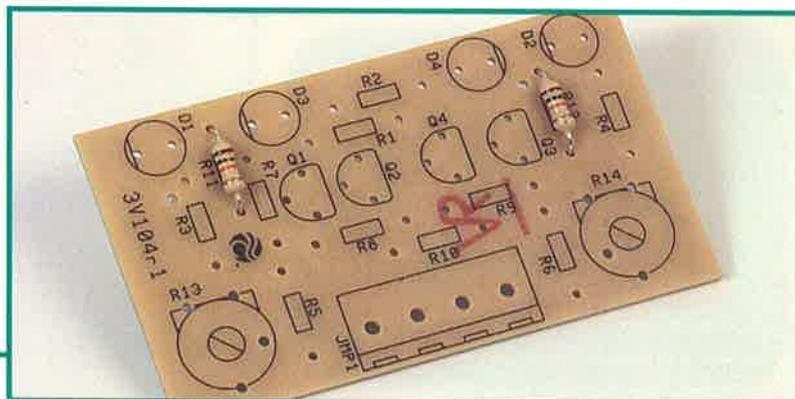
PL 88

Montaggio passo a passo



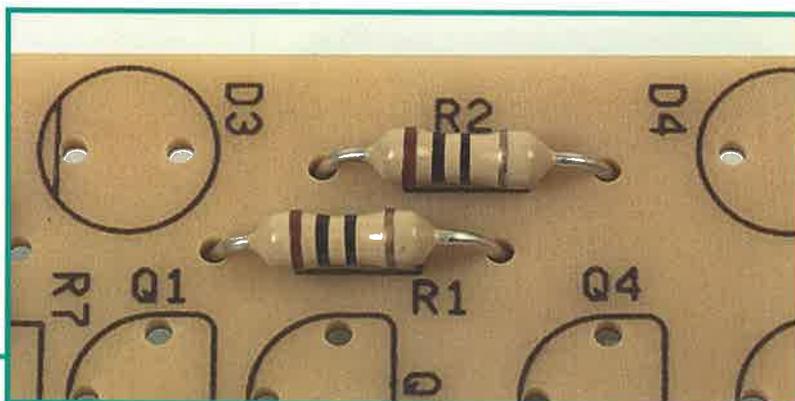
Questa scheda, di cui iniziamo ora il montaggio, conterrà due fototransistor e l'elettronica necessaria per implementare due rilevatori di luce. In questo modo Monty potrà reagire alle stimolazioni luminose.

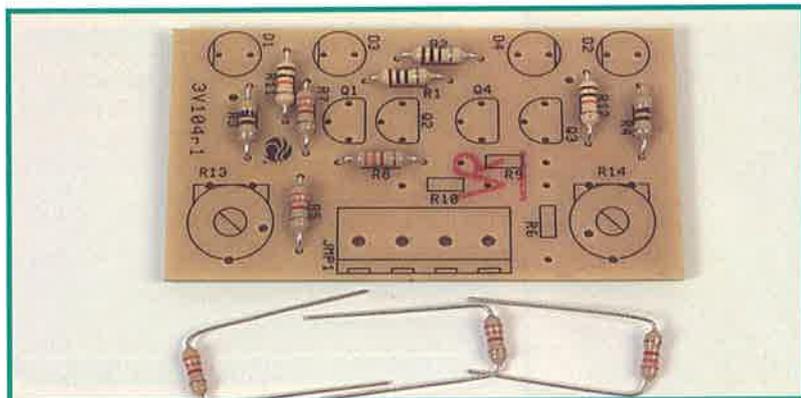
Per prima cosa salderemo due resistenze da $10\text{ K}\Omega$, le quali andranno inserite in R11 e R12. Il codice a colori di queste resistenze è marrone-nero-arancio.



Proseguiremo con due resistenze da $470\ \Omega$, il cui codice a colori è giallo-viola-marrone. Le loro posizioni sono R3 e R4. Come sempre, le resistenze si devono inserire sino a che rimangono a filo della scheda.

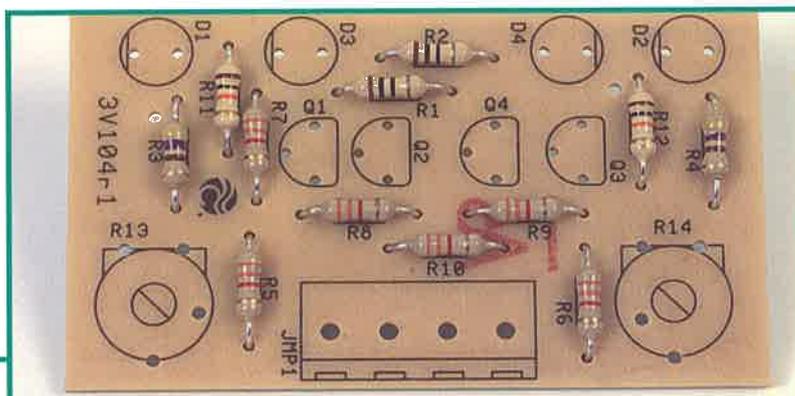
Dovremo saldare due resistenze da $10\ \Omega$, nelle posizioni R1 e R2. Le identificheremo utilizzando la tavola del codice a colori (marrone-nero-nero).



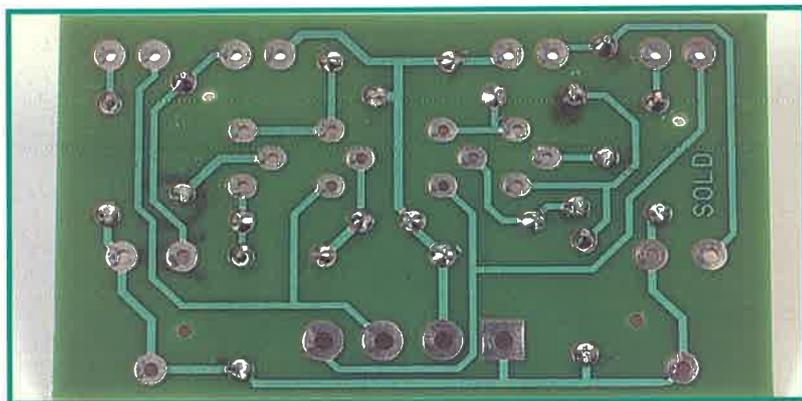


Salderemo ora sei resistenze da 3,3 K Ω . Le loro posizioni sono R5-R6-R7-R8-R9 e R10. Il codice a colori di queste resistenze è arancione-arancione-rosso. Faremo in modo di collocarle tutte con la banda dorata nello stesso verso, per questioni di estetica.

Questa è la scheda dei sensori di luce con i componenti che abbiamo saldato sino ad ora; abbiamo già saldato tutte le resistenze. Verificheremo che le loro posizioni corrispondano a quelle dell'immagine.

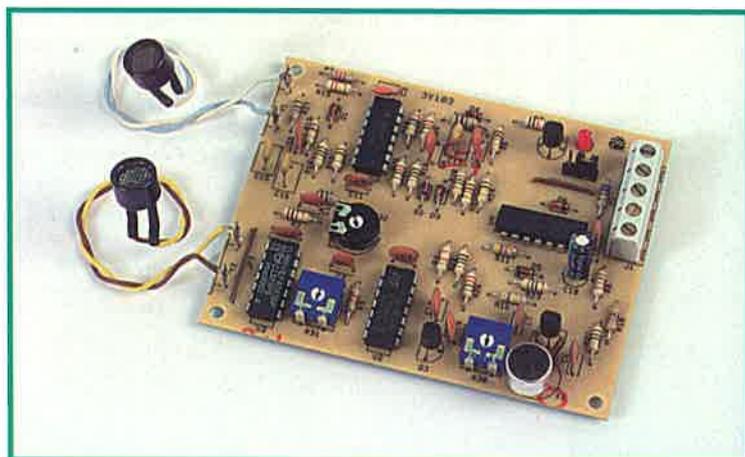


Le saldature delle resistenze devono avere questo aspetto: piccoli coni, non a forma di goccia, ma appoggiati sulla superficie. Con un tronchesino, o delle forbici, taglieremo la parte di reoforo in eccesso dopo la saldatura.



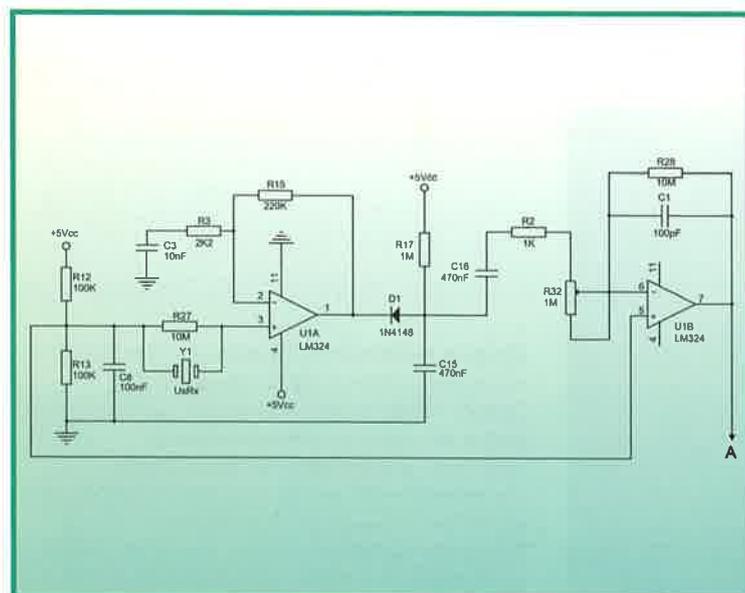
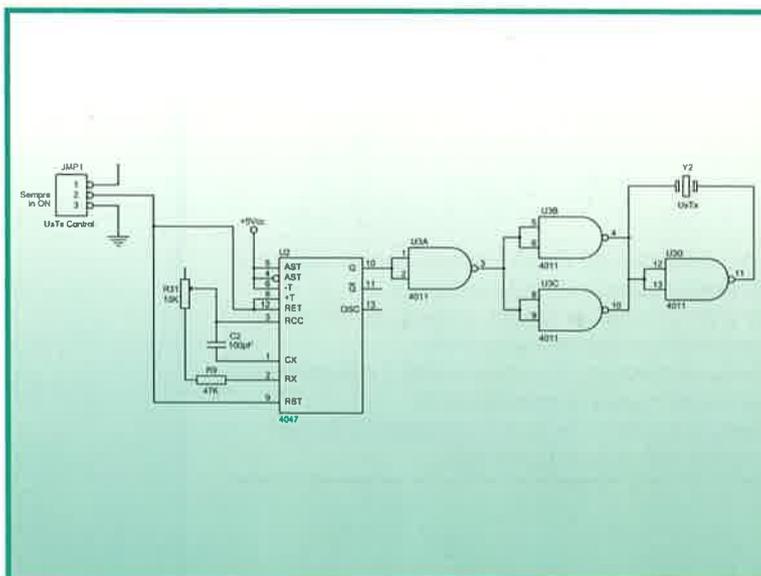
Questo è un pezzo della testa di Monty. Lo dobbiamo conservare per quando monteremo la struttura meccanica sulle schede elettroniche, per completare la costruzione del microrobot.



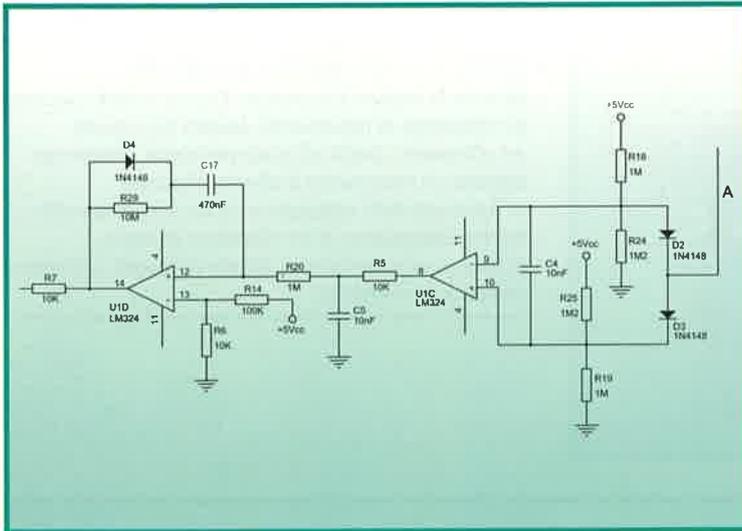


Dopo aver terminato il montaggio della scheda dei sensori, riportata in figura, passiamo a studiarne lo schema e il funzionamento, per ottenere da essa le migliori prestazioni. Questa scheda possiede un rilevatore di movimento, basato su capsule ad ultrasuoni, grazie al quale possiamo rilevare un oggetto in movimento a diverse distanze, in funzione della regolazione di un potenziometro. Inoltre disponiamo di un rilevatore di soglie di suono, che permette a Monty di reagire a fronte di stimoli sonori esterni.

Per il rilevatore di movimento abbiamo bisogno di un circuito che produca un'oscillazione nota, per attivare la capsula di trasmissione, e di un altro circuito che contenga la capsula di ricezione e ci indichi quando riceve qualcosa. Nella figura è riportato lo schema della sezione di trasmissione, che è un oscillatore, il cui periodo è controllato dal circuito 4047, e che produce un'onda a 40 kHz. Alla sua uscita abbiamo un quadruplo inverter formato da porte NAND che producono un effetto amplificatore.

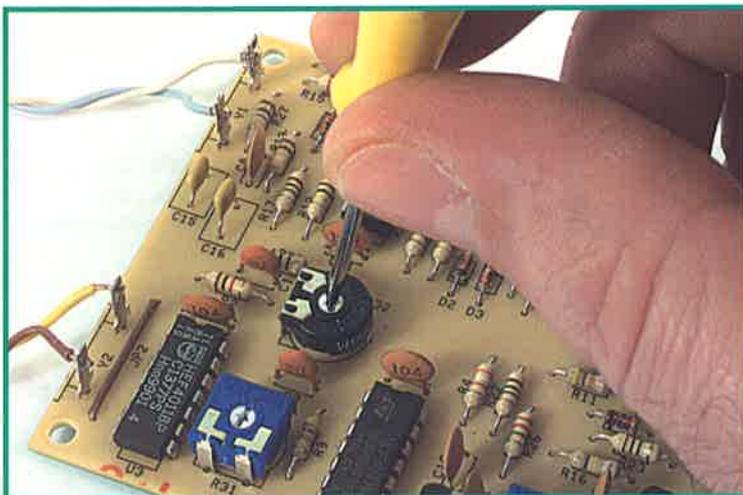
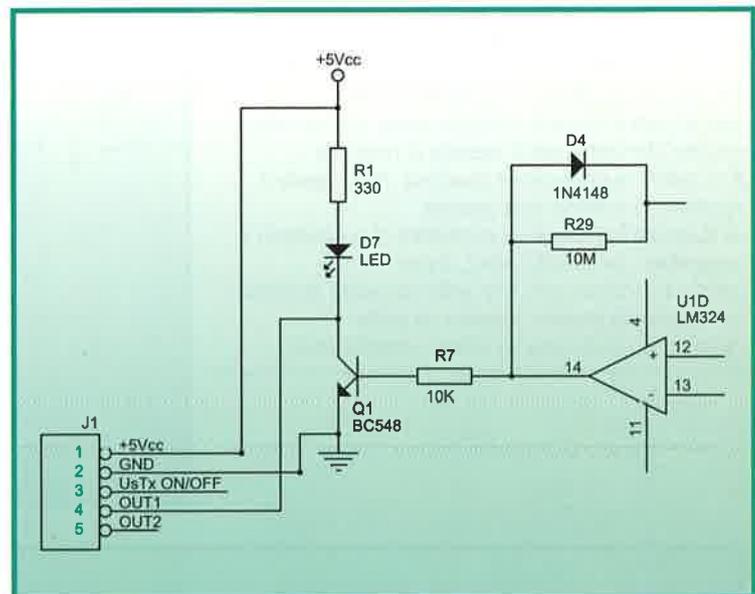


La sezione di ricezione è formata da quattro stadi, ognuno dei quali contiene un amplificatore operazionale LM324; nell'immagine sono mostrati i primi due stadi. Abbiamo una tensione di ingresso fissata da R12 e R13, la quale è modulata dalla capsula ricevente ad ultrasuoni, e amplificata dall'operazionale. A questo punto il segnale si recupera attraverso il diodo D1 e la resistenza R17, ed entra nel secondo stadio, nel quale torna ad essere amplificato, raggiungendo un livello di continua che rappresenta l'intensità della involvente.



Nella figura sono riportati gli ultimi due stadi della sezione di ricezione del rilevatore di movimento. Il terzo stadio è un rilevatore a finestra, per i segnali sopra e sotto la soglia di $\pm 0,7$ V fissata dai diodi D2 e D3. Quando il segnale supera questi valori significa che c'è movimento, viene polarizzato uno dei due diodi, e il segnale risultante è amplificato dall'operazionale. Il quarto stadio è un flip-flop monostabile con una costante di tempo fissata da R29 e C17.

Come risultato del circuito elettronico della sezione di ricezione, quando c'è del movimento si accenderà per un secondo il diodo LED D7, inoltre otterremo un segnale logico 0 che potremo applicare al microcontroller. Il collegamento si realizza tramite la morsettiere dei sensori J1 (OUT1), la quale è collegata a uno degli ingressi della scheda di potenza.

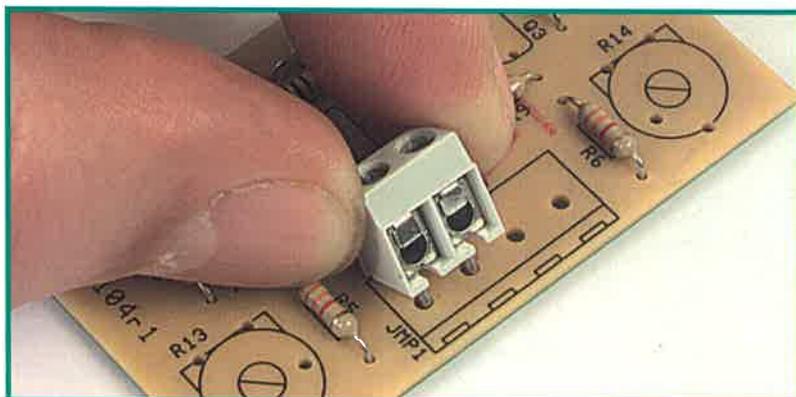


Potremo modificare la sensibilità del rilevatore di movimento variando il potenziometro R32. Quanto maggiore sarà la sensibilità, tanto maggiore sarà la distanza a cui potremo rilevare. A seconda dell'applicazione che vogliamo implementare con Monty, ci interesseranno distanze di rilevazione più lunghe o più corte. Le capsule a ultrasuoni devono essere collocate in modo parallelo e a una distanza di circa 4,5 cm.

MODULO DELLE LUCI

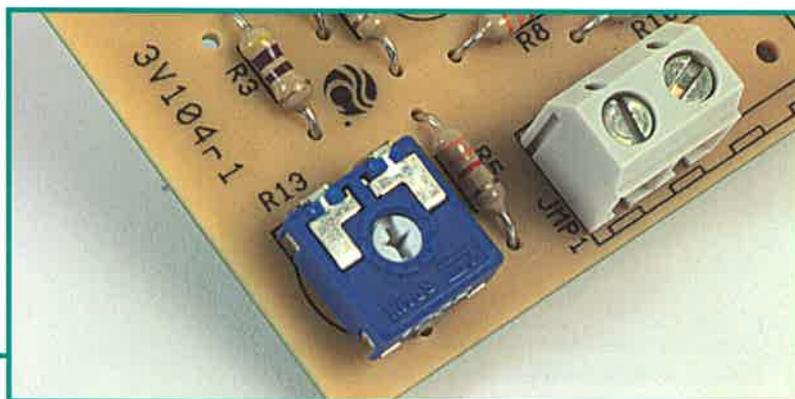
PL 90

Montaggio passo a passo

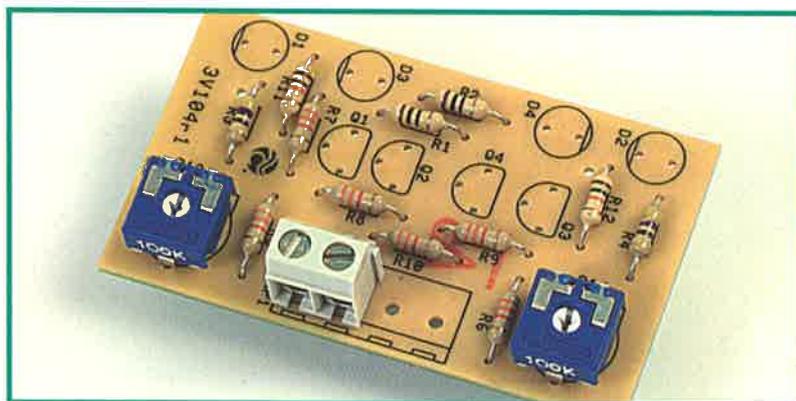


Montiamo una morsettiere a due vie in JMP1 (in seguito ne monteremo un'altra). Dobbiamo inserirla totalmente nella scheda.

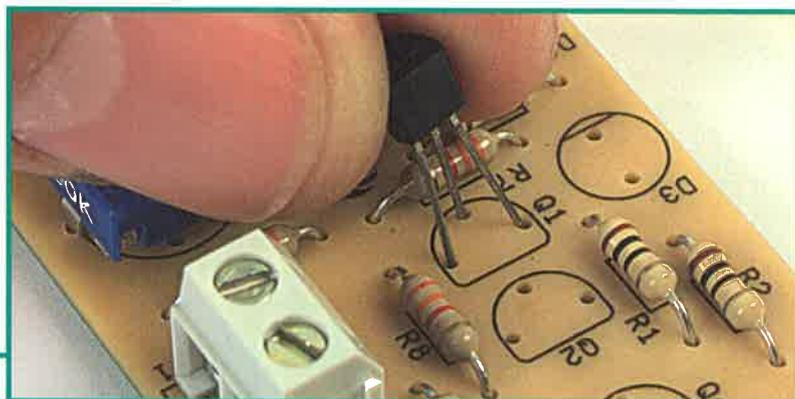
Dobbiamo montare e saldare un potenziometro da 100 K Ω in R13, nella posizione indicata nell'immagine. Lo dobbiamo introdurre completamente prima di realizzare le saldature.

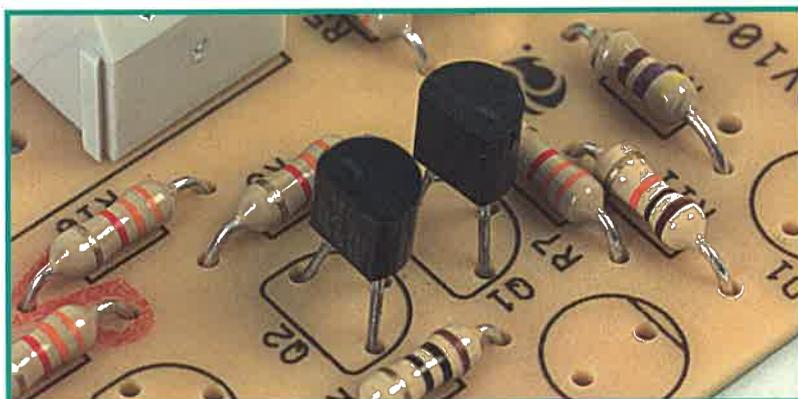


Ora salderemo un altro potenziometro da 100 K Ω in R14. Come quello in R13, deve rimanere a filo della scheda e le saldature dovranno essere a forma di cono, appoggiate alla superficie e con la giusta quantità di stagno.



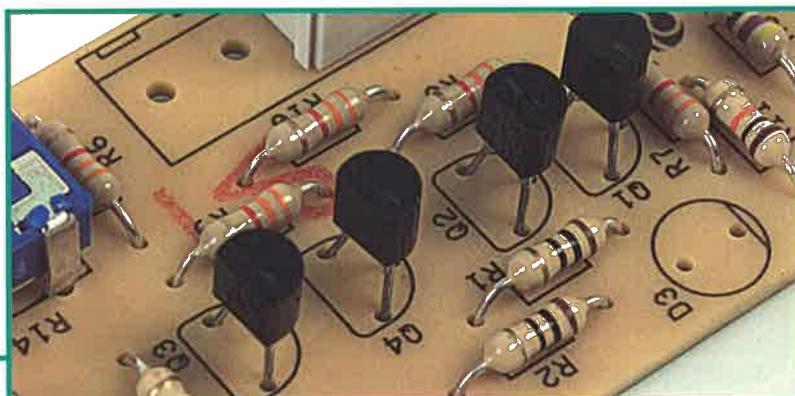
Salderemo un transistor modello BC548 nella posizione corrispondente a Q1, come mostrato nell'immagine. È necessario far coincidere la forma a mezzaluna del transistor con la forma serigrafata sulla scheda.



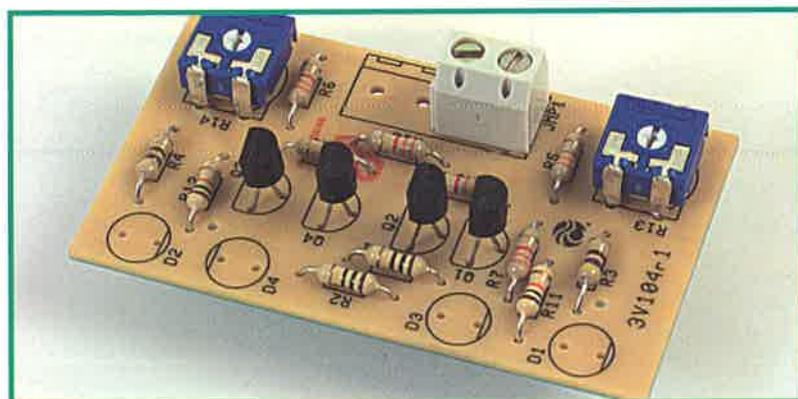


Dobbiamo saldare un altro transistor modello BC548 nella posizione Q2. I transistor devono rimanere sollevati qualche millimetro dalla superficie e non devono essere totalmente inseriti nella scheda.

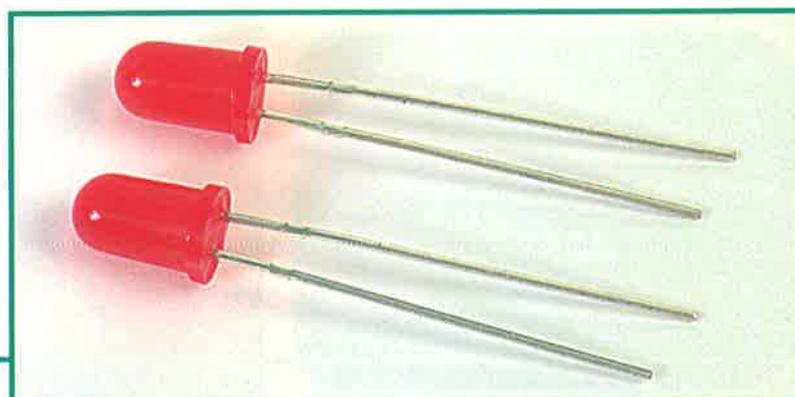
Seguendo le stesse precauzioni saldiamo gli ultimi due transistor BC548 di cui è composto il modulo dei sensori di luce. Questi transistor si collocano negli spazi segnati come Q3 e Q4.

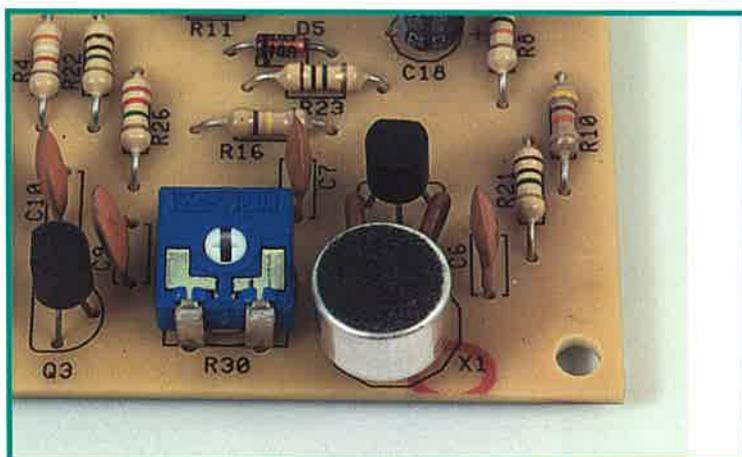


Ecco come deve rimanere la scheda dei sensori di luce con i componenti saldati sino a questo momento. Verificate la corretta ubicazione e l'orientamento dei diversi componenti.



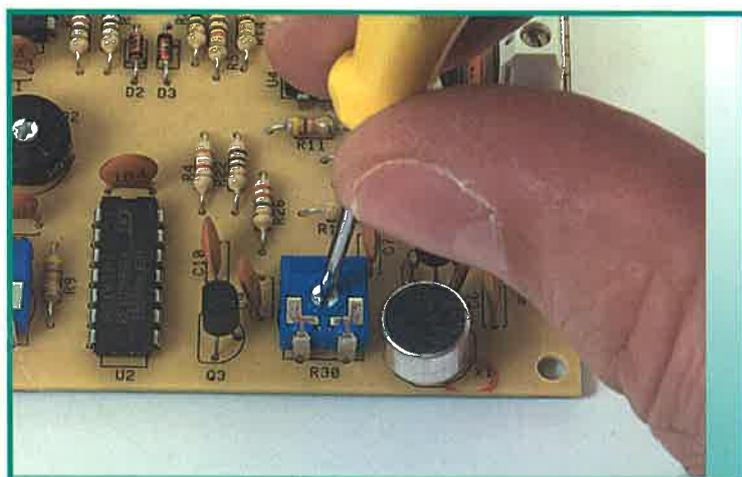
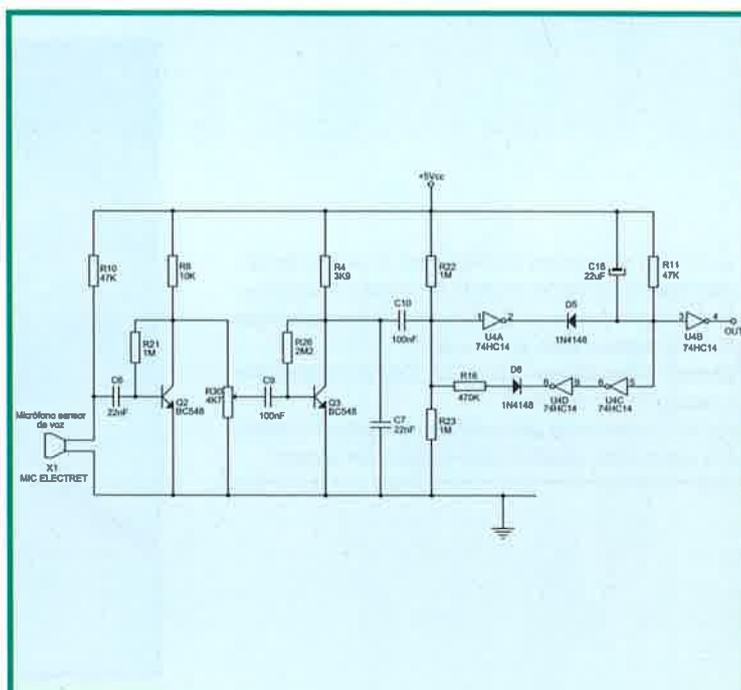
I diodi LED allegati al presente fascicolo devono essere conservati per il loro successivo montaggio. Andranno collocati, mediante faston, ai loro posti, contrassegnati come D1 e D2.



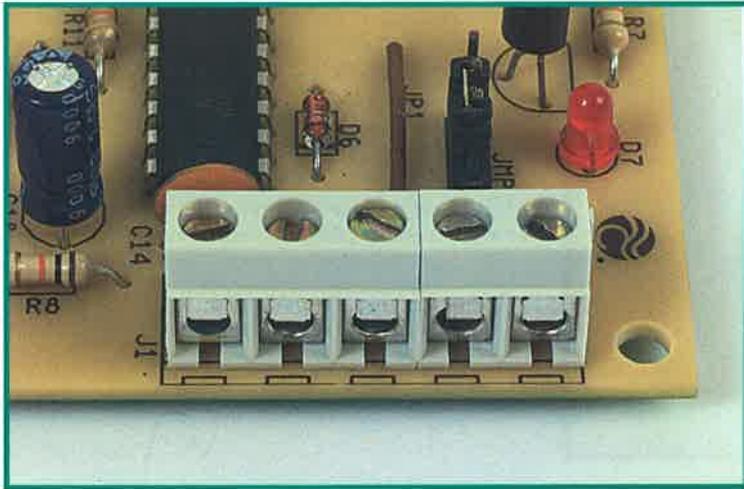


Nell'immagine è riportata una parte della scheda dei sensori impiegata per realizzare il rivelatore di suono. Questo sensore, basato su di un microfono, è molto utile per fare in modo che il robot reagisca agli stimoli esterni, e si possa programmare per realizzare determinati compiti quando battiamo le mani o emettiamo un suono. La soglia sonora a partire dalla quale si attiva il sensore è regolabile.

Questo è il circuito elettronico del rivelatore. Si basa su di un microfono che reagisce ai suoni e produce un segnale che, se ha sufficiente intensità, attiverà il resto dell'elettronica, e produrrà un 1 logico all'uscita. Gli elementi attivi del circuito sono due transistor BC548 che amplificano il segnale del microfono. Gli inverter "Trigger" contenuti nel 74HC14 sono incaricati di conformare il segnale logico che si ottiene sul piedino 5 della morsetteria J1 (OUT2).

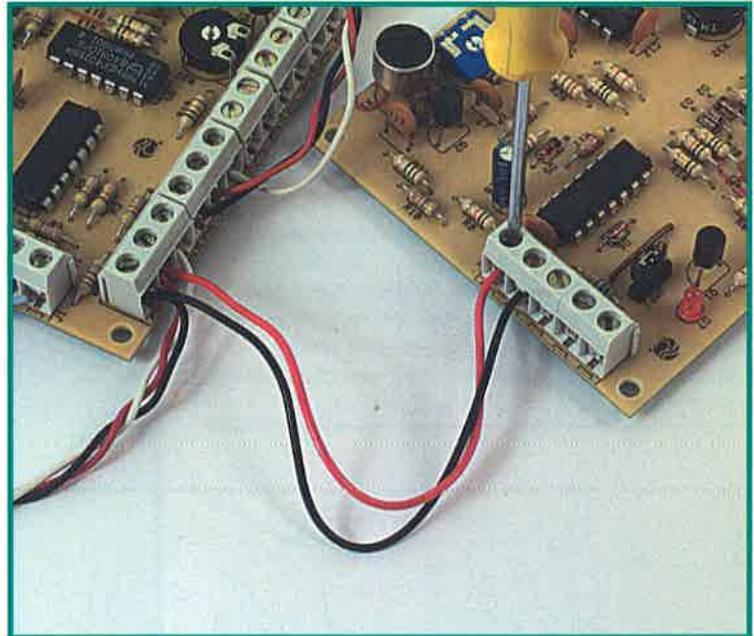


Utilizzeremo il potenziometro R30 per fissare il livello sonoro di attivazione del sensore. Se il robot andrà in un ambiente rumoroso, la soglia non potrà essere molto bassa, altrimenti il rivelatore sarebbe continuamente attivo. Invece, se programiamo Monty in modo che realizzi compiti di vigilanza, la soglia dovrà essere molto bassa, affinché riveli qualsiasi possibile intruso.

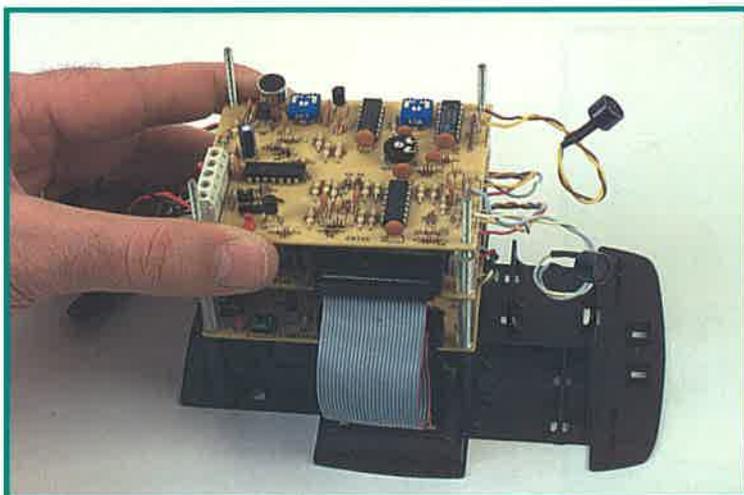


Connettore tramite il quale alimenteremo la scheda dei sensori. Così come è mostrato nell'immagine, sul primo ingresso di sinistra dobbiamo introdurre +5 V e nel secondo la massa. Con questi due segnali la scheda rimarrà alimentata. Nel terzo ingresso possiamo introdurre il segnale del microcontroller che serve per attivare o disattivare la sezione di trasmissione del rivelatore ad ultrasuoni di movimento. Nel quarto ingresso abbiamo l'uscita del rivelatore di movimento e nel quinto l'uscita del rivelatore di suono.

La scheda dei sensori si alimenterà e sarà collegata con il microcontroller tramite la scheda di potenza. Dalla scheda di potenza preleveremo i 5 V necessari per l'alimentazione della scheda dei sensori. Inoltre disponiamo di una serie di morsetti che contengono linee libere del PIC che potremo utilizzare per collegare gli ingressi e le uscite dei rivelatori della scheda dei sensori.



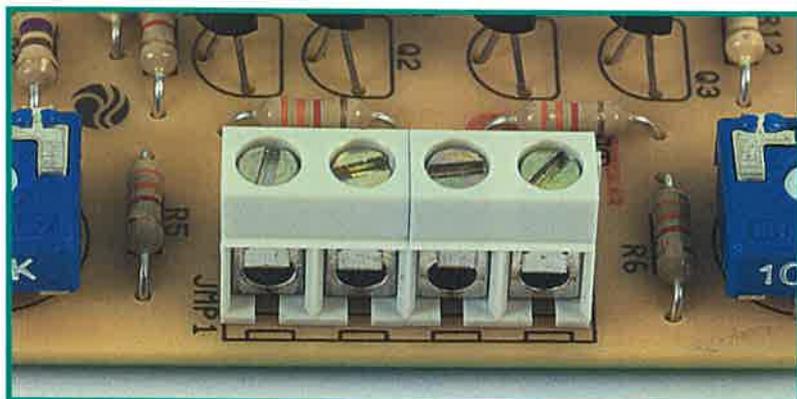
Oltre al collegamento della scheda di potenza con quella dei sensori, sarà necessario che la scheda di controllo sia a sua volta collegata con quella di potenza tramite il connettore Pic-Bus, tramite il quale i segnali arrivano sino al microcontroller, che è il cervello del microrobot. In questo modo avremo una struttura a torre come riportato nell'immagine. Per fare questo inseriremo le schede nelle barre filettate del telaio e le separeremo mediante pezzi di tubo di PVC che taglieremo a 16 mm. Possiamo lasciare sempre collegate le schede, anche se per programmare dobbiamo rispettare le posizioni dell'interruttore PROG/RUN della scheda di controllo.



MODULO DELLE LUCI

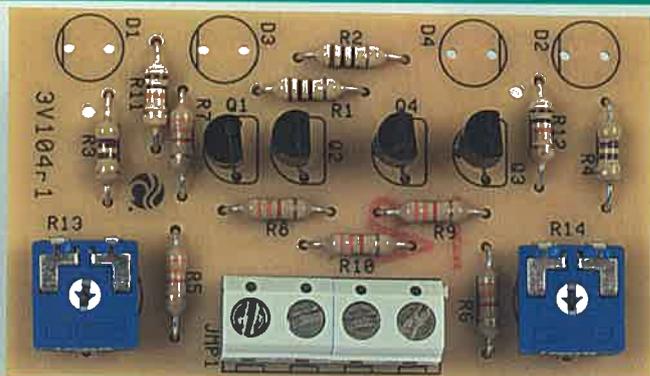
PL 92

Montaggio passo a passo



In JMP1 monteremo la morsettiere a due contatti che mancava, così come si vede nella figura. Abbiamo già le quattro morsettiere necessarie per collegare il modulo di sensori di luce con la scheda di potenza.

Questo fototransistor servirà per rilevare il livello di luce. La sua giunzione base/collettore si polarizza in funzione della quantità di luce incidente. Con il circuito appropriato otterremo dei livelli logici che saranno processati tramite il "cervello" di Monty.



Nell'immagine vediamo l'aspetto attuale della scheda che contiene i circuiti sensori di luce. La scheda è totalmente montata, mancano solo i due diodi LED, e i due fototransistor. Questi componenti saranno inseriti nella testa di Monty e si collegheranno alla scheda mediante dei fili.

Prepareremo otto pezzi di tubetto termorestringente. Serviranno per isolare fra loro i piedini dei LED e il fototransistor. I faston offrono una soluzione valida per saldare cavi su un circuito stampato, dato che si inseriscono sulla scheda (probabilmente sarà necessario ingrandire i fori con l'utilizzo di una punta), e i cavi si saldano su essi.

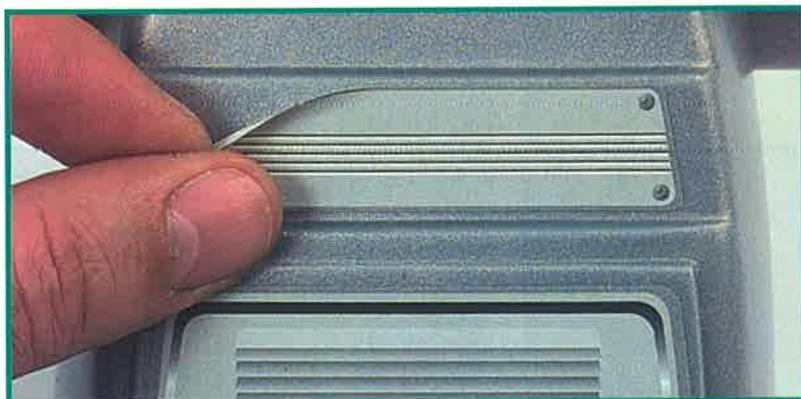




L'immagine riporta un foglio di dimensione A4 che contiene l'insieme degli adesivi decorativi del corpo di Monty. Prima di montarli si raccomanda di pulire completamente il corpo di Monty, in modo che aderiscano perfettamente.

Inizieremo, come si vede nell'immagine, aprendo la finestra predisposta, che dà accesso al connettore DB25 utilizzato per la connessione col PC. Nella parte interna del corpo, con l'aiuto di un piccolo cacciavite o di un elemento da taglio, facciamo pressione sui segni di rottura, finché si libera questa finestra.

Con una lima o qualcosa di simile elimineremo i possibili trucioli eventualmente rimasti

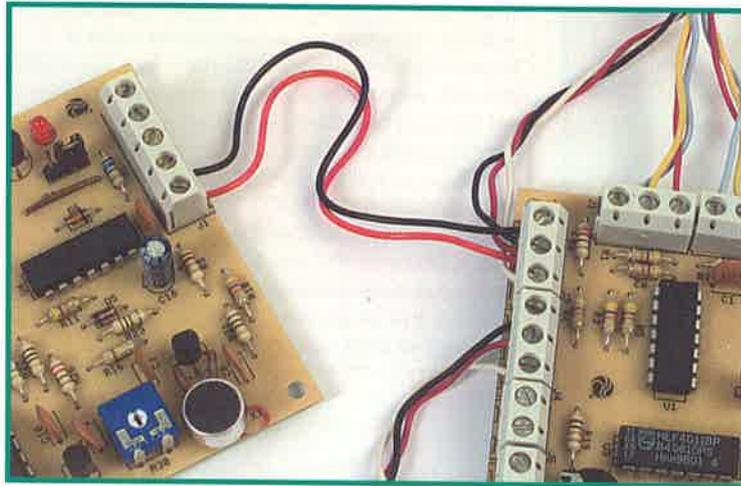


Inizieremo a decorare la prima parte del corpo di Monty, che è stata fornita in precedenza, e che forma la parte posteriore.

Gli adesivi si trovano appiccicati sulla lamina. Si raccomanda di staccarli con cura in modo che non si rompano. Applicandoli faremo attenzione ad allinearli correttamente per ottenere un buon risultato estetico.

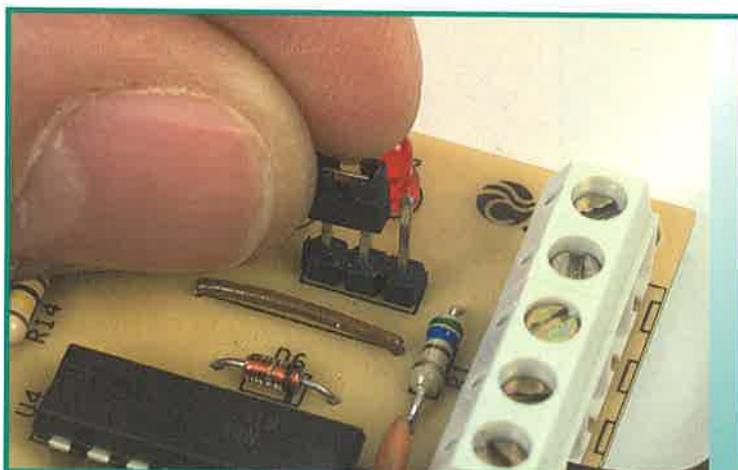
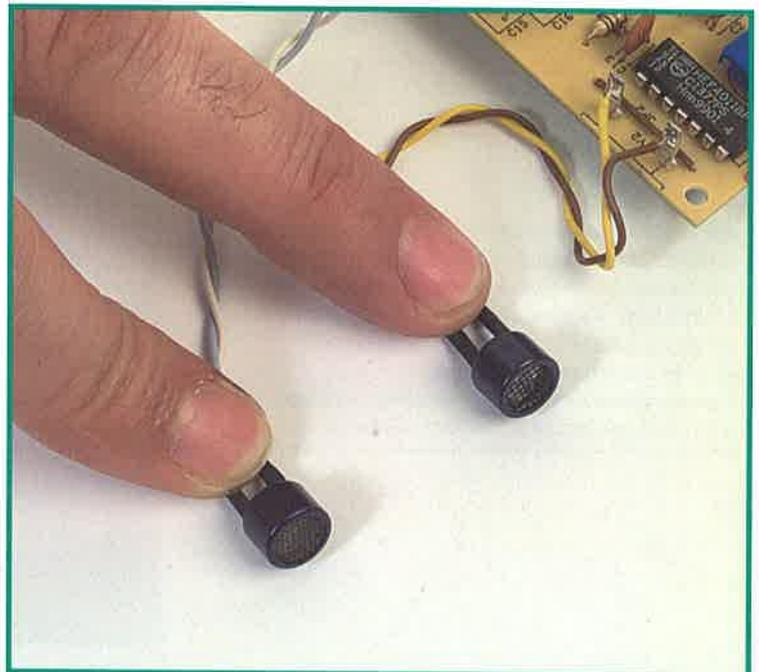
L'immagine ci offre un'idea di come devono essere collocati i diversi adesivi che decorano la parte posteriore del corpo di Monty. Fra poco decoreremo la parte anteriore in modo simile.



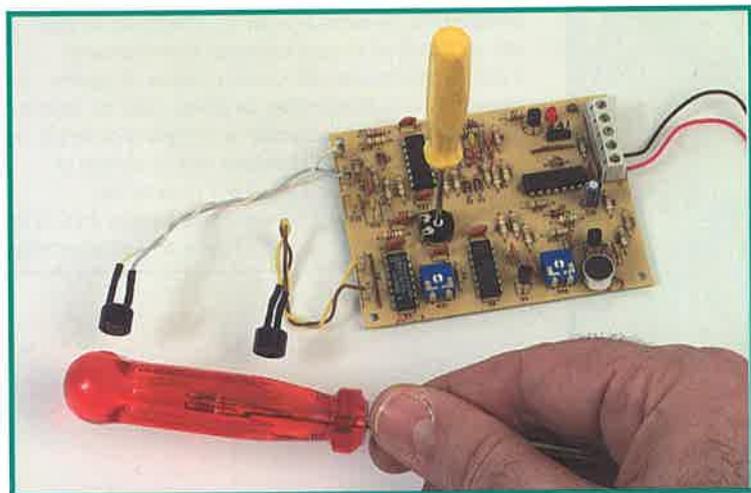


Realizziamo un esercizio con la scheda dei sensori per verificare se è stata montata correttamente e per familiarizzare con i sensori prima di gestire i loro segnali dal microcontroller. La prima cosa da fare sarà alimentare la scheda tramite la scheda di potenza, la quale a sua volta sarà collegata con la scheda di controllo. Gli ingressi 1 (5 V) e 2 (massa) del connettore J1 si collegheranno con l'uscita 2 (5 V) e 1 (massa) della morsettieria J3 della scheda di potenza.

Le capsule ad ultrasuoni devono essere perfettamente allineate con una distanza minima fra loro, in modo che non producano interferenze. La capsula di emissione invia un segnale che dopo aver rimbalzato su un ostacolo torna alla capsula di ricezione. Se l'allineamento non è adeguato e la distanza fra le capsule diversa da 5 centimetri, il sensore non si comporterà correttamente.

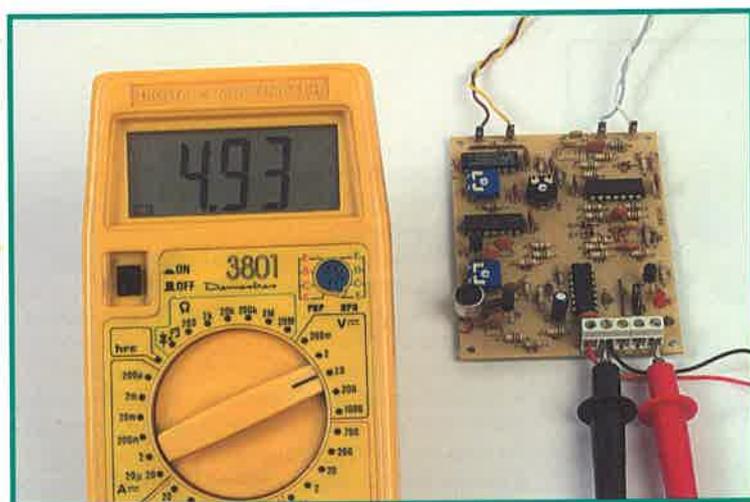
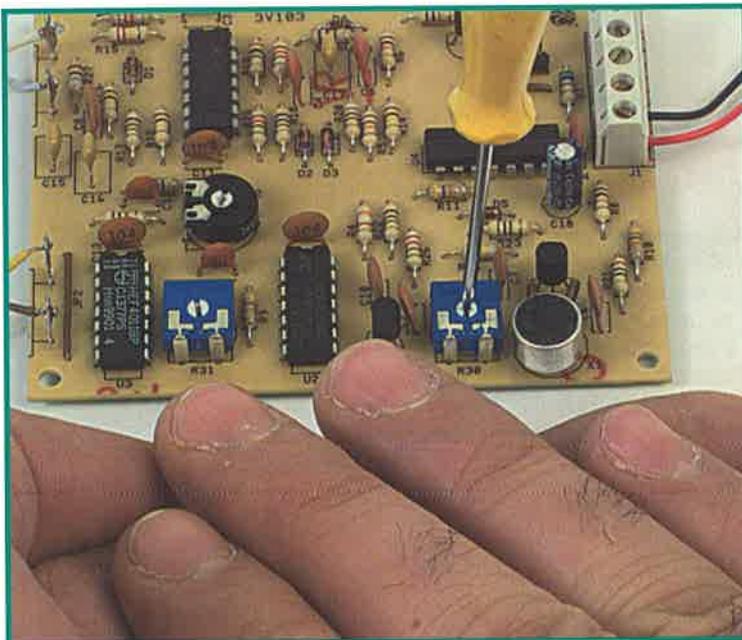


Con il jumper JMP1 selezioneremo se la sezione di trasmissione del rilevatore di movimento deve essere sempre attiva, o se la sua attivazione o disattivazione sono seguite dal microcontroller. Che l'attivazione sia controllata può essere interessante per quelle applicazioni in cui il robot si sta muovendo in continuazione, e a un determinato momento si ferma per rilevare un movimento. In questo istante attiveremo la sezione di emissione. Comunque, per realizzare questo esercizio dobbiamo lavorare con l'emettitore sempre attivo, ponendo il jumper JMP1 a massa, come si vede nell'immagine.



Ora passeremo la mano o un oggetto davanti ai sensori ad ultrasuoni e osserveremo come si illumina il diodo D7 ogni volta che c'è un movimento. Dovremo modificare la posizione del potenziometro per variare la sensibilità del rilevatore e studiarne il comportamento. Ogni volta che si illumina il LED, si invia un segnale logico "0" alla connessione 4 della morsettiera J1 (OUT1), e quindi arriverà fino al microcontroller. Inoltre dobbiamo regolare il potenziometro R31 sino a che il comportamento del sensore sarà totalmente soddisfacente. Questo è il potenziometro che regola la vibrazione della capsula ad ultrasuoni a 40 KHz.

Per verificare la rilevazione del suono dobbiamo mantenere la tensione di alimentazione all'ingresso del connettore J1 tramite le morsettiere. Modificando il valore del potenziometro R30 variamo la sensibilità del rilevatore. Batteremo le mani per provare il rilevatore. Con maggiore sensibilità, il rilevatore si attiverà con suoni più bassi e a maggiore distanza.

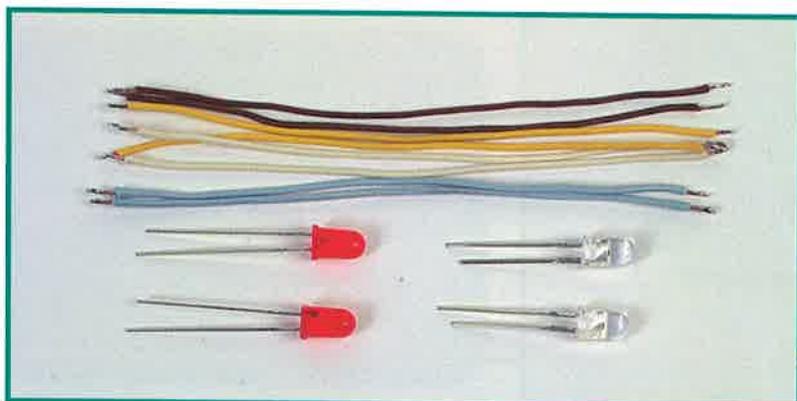


Per verificare come si attiva il rilevatore possiamo usare un tester. Lo regoleremo per la misura di tensione con le morsettiere, come si vede nell'immagine; il puntale nero a massa, e quello rosso al collegamento 5 della morsettiera J1 (OUT2). Ogni volta che il rilevatore si attiva, sul tester visualizzeremo 5 V. Quando è a riposo 0 V. Questo sarà il segnale che arriverà al microcontroller, quando collegheremo questa morsettiera alla scheda di potenza.

MODULO DELLE LUCI

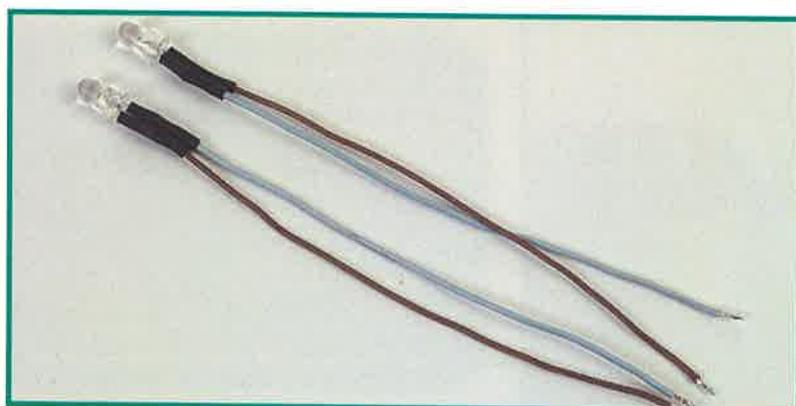
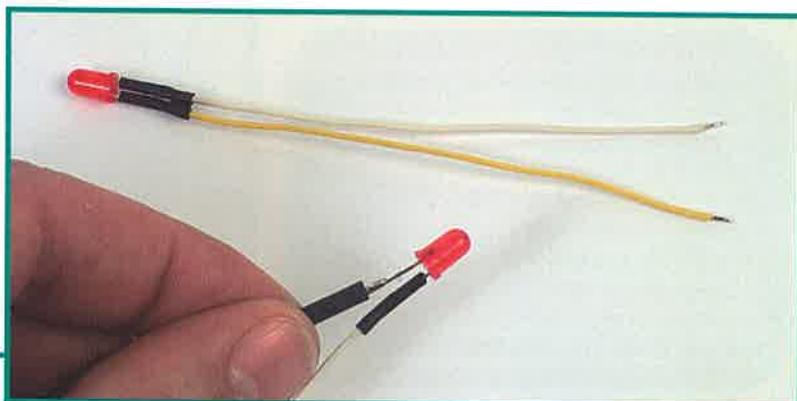
PL 94

Montaggio passo a passo



Prepariamo i cavi che servono per collegare i diodi LED e i fototransistor con la scheda dei sensori di luci. Avremo bisogno di otto pezzi da 10 centimetri ognuno: due di colore bianco, due gialli, due azzurri e gli altri due di colore marrone. Peleremo per 5 mm gli estremi di ogni cavo, li intrecceremo e prestagneremo, perché non rimangano reofori liberi.

Salderemo i cavi ai diodi LED rossi. Il piedino corto del LED corrisponde al catodo, e su di esso salderemo i due cavi gialli. I piedini lunghi sono dell'anodo e dovremo saldare su di essi i cavi bianchi. Ogni piedino deve essere protetto mediante pezzi di termorestringente, come illustrato nella fotografia. In questo modo eviteremo che si tocchino fra loro, nel caso si piegassero.



Ora salderemo i cavi dei due fototransistor. I piedini corti corrispondono al collettore, e su di essi salderemo i cavi di colore marrone. I piedini lunghi corrispondono all'emettitore e su ognuno di essi salderemo un cavo di colore azzurro. Come abbiamo fatto con i LED, proteggeremo ogni piedino con i corrispondenti pezzi di tubetto termorestringente.

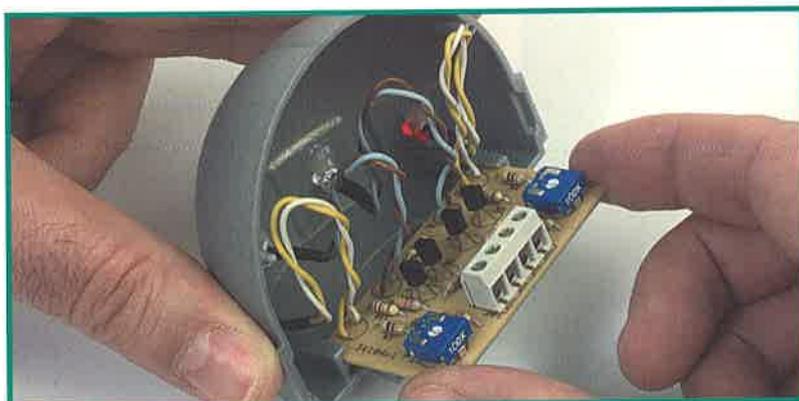
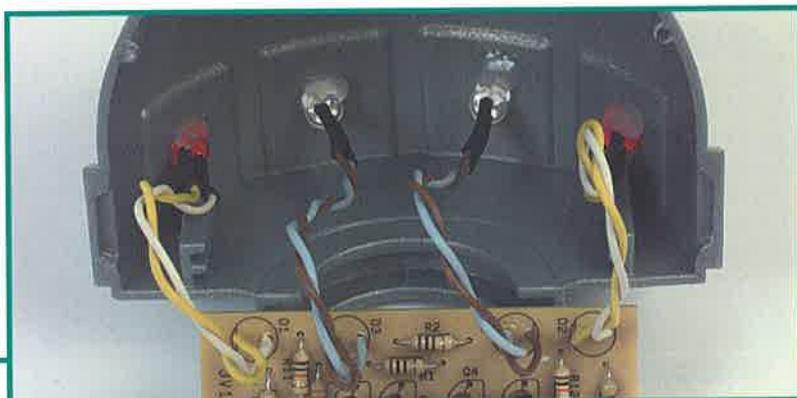
Prepareremo una delle metà della testa di Monty, che considereremo come la parte anteriore o "faccia" del robot. Inizieremo, come si vede nella fotografia, aprendo i fori predisposti nella parte interna della testa con l'aiuto di un cacciavite, o simile, faremo pressione sui segni di rottura fino a perforare. Con una lima elimineremo i trucioli che potrebbero rimanere.





I due diodi LED dovranno essere introdotti sui due fori estremi della testa. Saranno fissati mediante adesivo. Guardando la testa frontalmente, sul lato interno, i cavi del LED di sinistra si saldano sulla scheda, nel punto riferito come D1. Il LED di destra in D2. I cavi gialli corrispondono al catodo e devono essere introdotti nel foro orientato verso la riga della serigrafia.

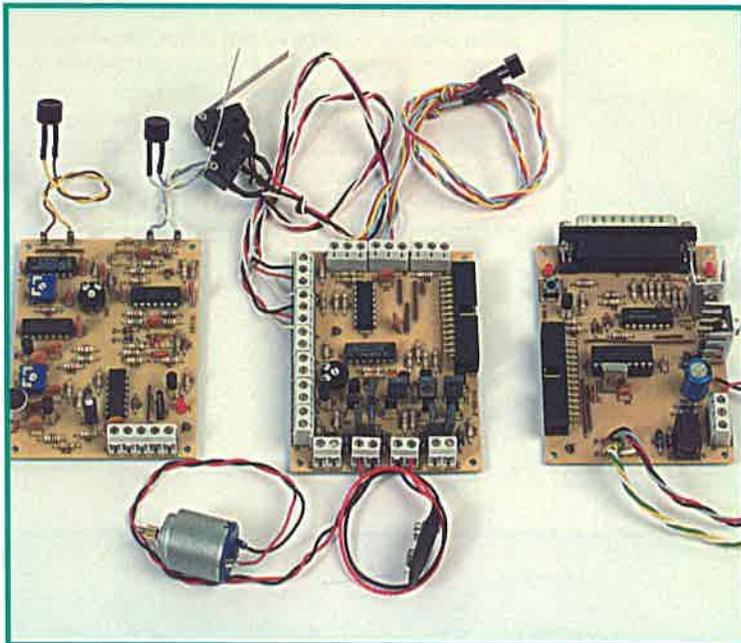
Monteremo ora i due fototransistor che si introducono per i due fori centrali della testa. Possono essere fissati mediante qualsiasi tipo di adesivo. Guardando la testa in modo frontale, sul lato interno, i cavi del fototransistor di sinistra si salderanno nel punto indicato come D3, e quello di destra in D4. I cavi marrone corrispondono al collettore e saranno inseriti e saldati nel foro orientato verso la riga della serigrafia.



Una volta realizzata la connessione precedente, lo stampato viene introdotto nelle guide interne della testa. I cavi devono essere piegati con cura perché rimangano alloggiati in questa metà della testa di Monty e non diano fastidio in seguito.

Per completare la testa di Monty, procederemo a decorare, con gli adesivi, le due metà che la compongono, così come possiamo vedere nell'immagine. A partire da questo momento siamo pronti per collegare la testa con la scheda di potenza.





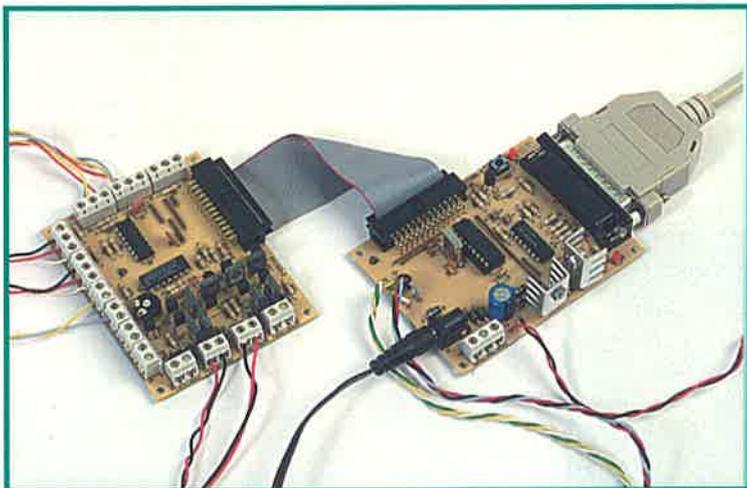
In questo esercizio controlleremo la scheda dei sensori tramite il microcontroller. Ogni volta che utilizziamo la scheda dei sensori utilizzeremo anche la scheda di potenza, dato che la prima si collega tramite questa. A sua volta la scheda di potenza si collega mediante il Pic-bus alla scheda di controllo dove si trova il microcontroller.

Il programma consiste nel rivelare il movimento e il livello del suono, in modo che entrambi gli eventi servano a far eseguire delle azioni al microcontroller. Abbiamo un motore collegato alla scheda di potenza, il quale rimarrà inattivo fino a che i sensori ad ultrasuoni rileveranno un movimento. A questo punto il motore si metterà in marcia, e non si fermerà fino a quando il microfono non rileverà un suono. Nell'immagine vediamo la configurazione classica delle porte di I/O.

```
c:\progra~1\mplab\asc_48\eserci.asm
1
2
3
4
5
6
7
8
9 INIZIO:
10
11
12
13
14
15
List p=16F84A
include "P16F84A.INC"
org 0x00
goto INIZIO
org 0x05
bsf STATUS,RP0
clrf PORTA
movlw b'00110000'
movwf PORTB
bcf STATUS,RP0
clrf PORTA
```

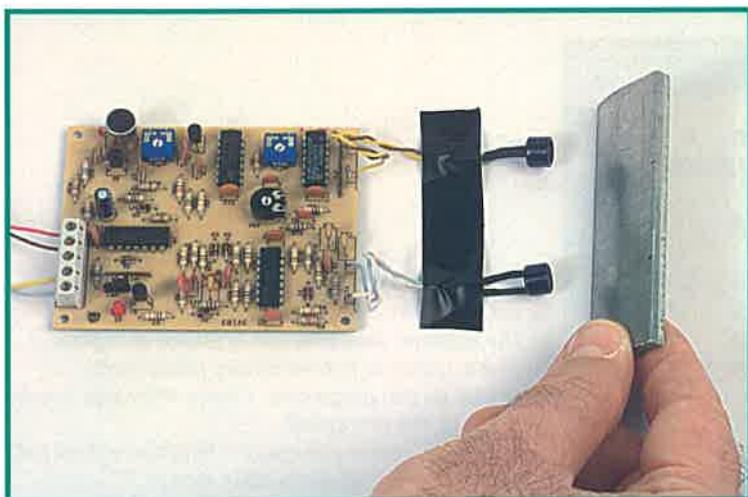
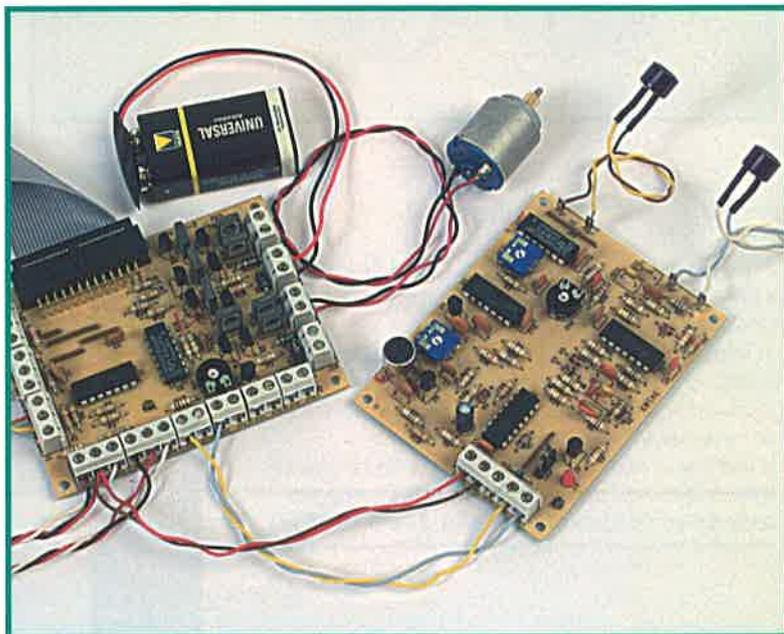
```
c:\progra~1\mplab\asc_48\eserci.asm
15
16 MOVIMENTO:
17
18
19
20
21 SUONO:
22
23
24
25
26
btfsc PORTB, 4
goto MOVIMENTO
movlw 0x01
movwf PORTA
btfss PORTB, 5
goto SUONO
clrf PORTA
goto MOVIMENTO
END
```

Questa è la parte principale del programma, come si può osservare è simile ai programmi che abbiamo realizzato con la scheda di ingressi e uscite, quindi lo tratteremo allo stesso modo di tutti gli altri programmi. L'unica differenza consiste nel fatto che gli ingressi e le uscite, invece che essere interruttori e diodi LED, saranno sensori di diversi tipi e motori.



Dopo aver scritto il programma e averlo compilato senza errori, procederemo alla programmazione del microcontroller. Per fare questo utilizzeremo il cavo parallelo e collegheremo la scheda di controllo al computer. L'interruttore della scheda di controllo deve essere in posizione PROG. Una volta programmato il chip, collegheremo la scheda di controllo a quella di potenza tramite il Pic-bus.

Dobbiamo collegare un motore alla morsettiera J11 della scheda di potenza (motore 1), e fornire l'alimentazione al medesimo mediante la morsettiera J13. Collegheremo la scheda di potenza con quella dei sensori tramite i cavi e i morsetti, per cui utilizzeremo due cavi per l'alimentazione e altri due per ottenere i segnali dei sensori. L'uscita OUT1 degli ultrasuoni (collegamento 4 della morsettiera J1) deve essere unita con la RB4 (collegamento 1 della morsettiera J6 della scheda di potenza). Da parte sua l'uscita OUT2 del microfono (connessione 5 della morsettiera J1) si porterà a RB5 (collegamento 1 della morsettiera J7 della scheda di potenza).



Possiamo provare il programma fissando le due capsule ad ultrasuoni in una posizione determinata. In seguito passeremo un oggetto davanti al rilevatore ad ultrasuoni e il motore si metterà in marcia. Dopo di che genereremo un suono che superi la soglia del microfono e osserveremo che il motore si fermerà. Possiamo modificare la sensibilità di entrambi i rilevatori, muovendo i potenziometri R30 e R31 della scheda dei sensori.

MODULO DEI SENSORI

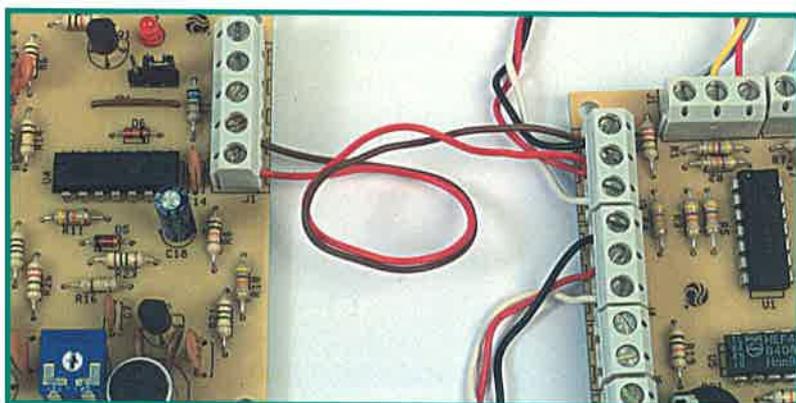
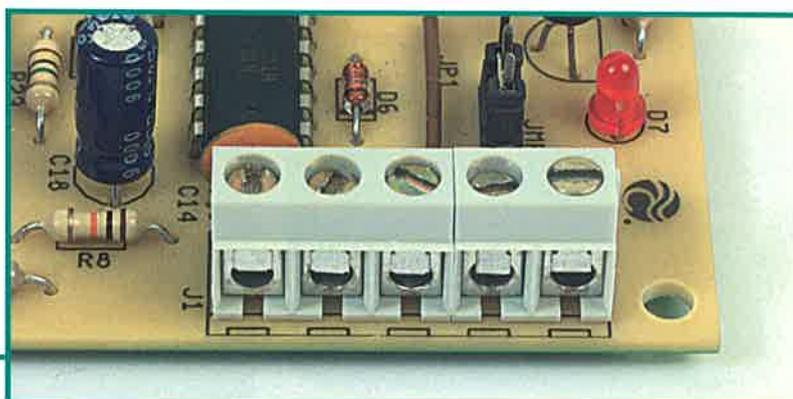
PL 96

Montaggio passo a passo



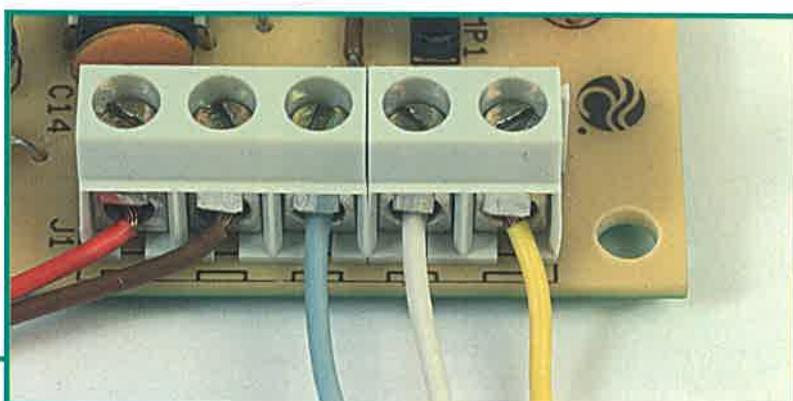
Utilizzeremo questi cinque cavi per realizzare le connessioni tra la scheda dei sensori e quella di potenza. La scheda dei sensori non si collegherà pertanto tramite il Pic-bus, ma mediante le morsettiere e dei cavi.

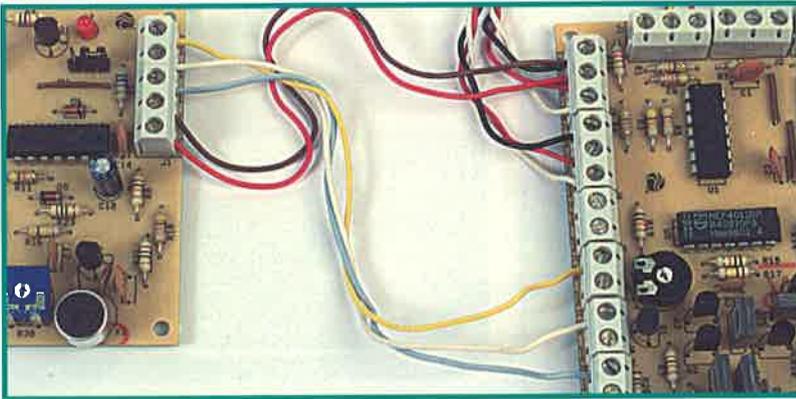
Questi sono i cinque morsetti che abbiamo a disposizione per fornire alimentazione tramite la scheda di potenza, e anche i segnali di controllo e quelli di uscita dei sensori. La scheda di potenza sarà quella che collega tutto con il microcontroller, tramite il Pic-bus.



L'alimentazione della scheda avviene tramite i morsetti 1 e 2, come si vede nell'immagine. Sul morsetto 1 dobbiamo portare 5 V e sul 2 la massa. Possiamo prendere questa alimentazione da qualsiasi punto della scheda di potenza in cui ci sia alimentazione; ad esempio dai morsetti 1 e 2 della morsettieria J3, come si vede nell'immagine.

I morsetti 3 e 4 servono per il controllo del sensore ad ultrasuoni. Il morsetto 3 ci permette di dare l'ordine di emissione degli ultrasuoni quando si pone a livello logico "0", e nel morsetto 4 (OUT1) abbiamo uno "0" logico ogni volta che si rileva un movimento. La funzione della morsettieria 5 (OUT2) è quella di ottenere il segnale del rilevatore di suono, che genera un "1" ogni volta che si supera la soglia sonora.





Possiamo collegare questi segnali di controllo al terminale 1 dei morsetti J6, J7, J8 e J9 della scheda di potenza, in cui abbiamo liberi i segnali RB4, RB5, RB6 e RB7 del PIC. I collegamenti definitivi delle schede rimangono come mostrato nell'immagine.

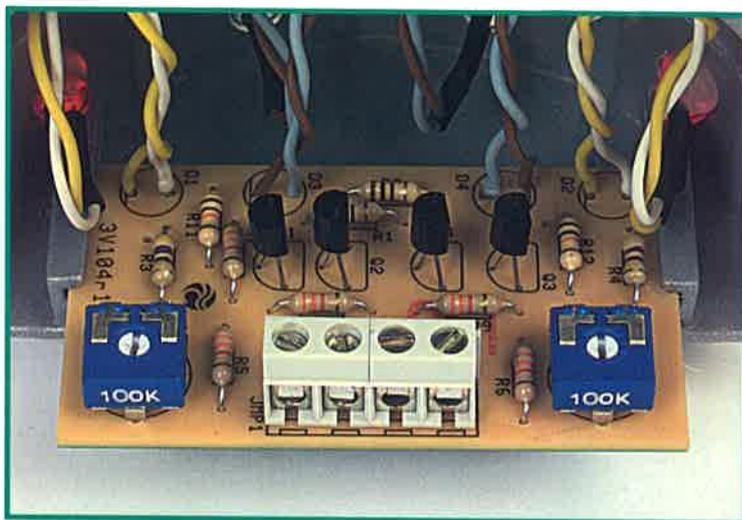
Questo è il secondo pezzo del corpo del robot. Corrisponde alla metà anteriore. Nell'immagine possiamo vedere l'apertura delle due sedi predisposte che permetteranno di collocare la capsula ad ultrasuoni. Utilizzeremo un piccolo cacciavite o un elemento di taglio con cui faremo pressione sui segni di rottura dalla parte interna del corpo. Con una lima elimineremo i possibili trucioli.



Decoreremo la seconda metà del corpo di Monty. Gli adesivi devono essere disposti con cura, in modo che non si danneggino, facendo attenzione ad incastrarli e allinearli correttamente, evitando le bolle d'aria.

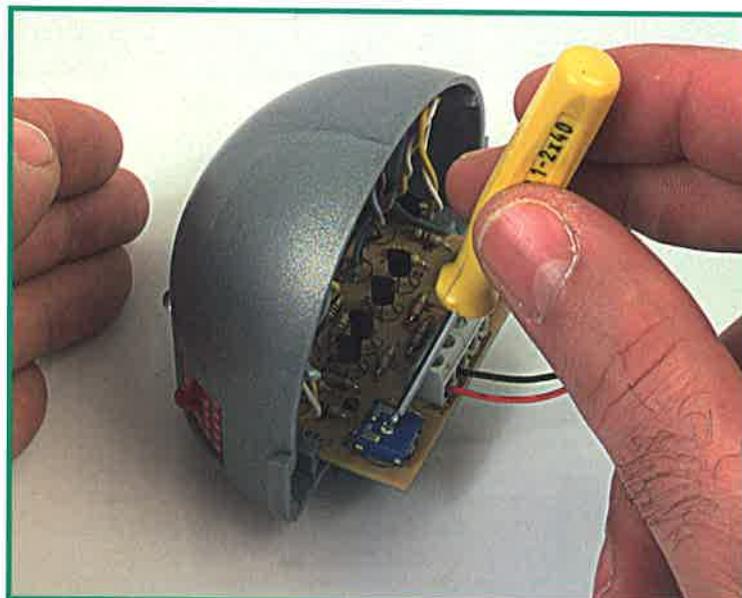
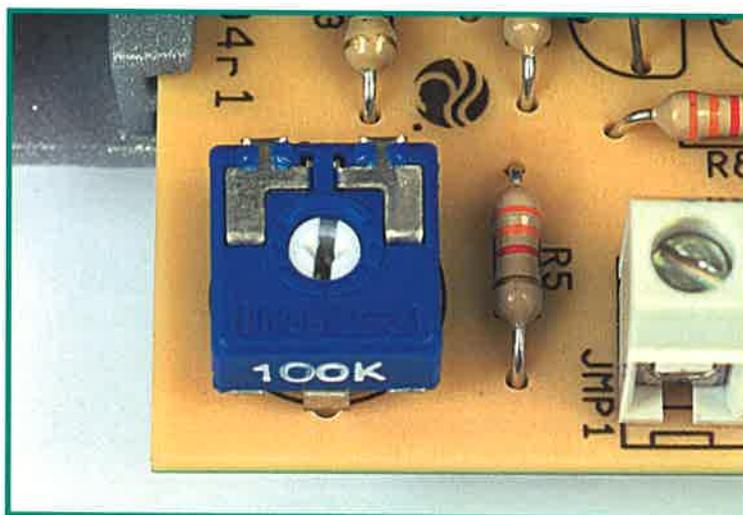
Queste immagini ci danno un'idea di come rimarrà decorata la parte anteriore del corpo di Monty. Da questo momento, quando lo riterremo opportuno, potremo assemblare le due parti assieme alla struttura portante e inserire l'elettronica all'interno di Monty.



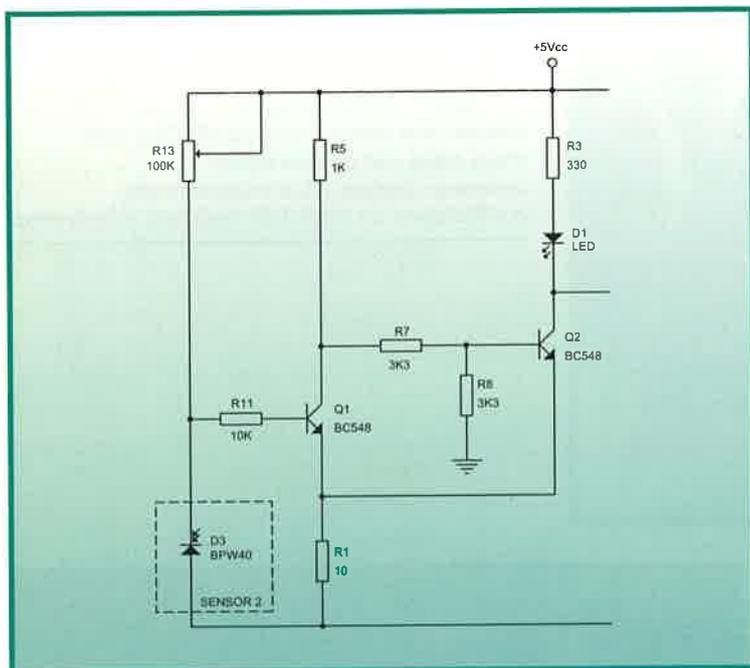


Questa è la scheda dei sensori di luce, già montata e preparata nei fascicoli precedenti, per funzionare all'interno del sistema di controllo di Monty. La scheda è stata progettata per rilevare i livelli di luce tramite i due sensori indipendenti. Ogni volta che si attiva uno dei due sensori, arriverà un segnale TTL al microcontroller, e si illuminerà un diodo LED rosso sulla scheda stessa.

I potenziometri R13 e R14 serviranno per regolare la sensibilità del Robot alla luce che riceve. In questo modo, in funzione dell'applicazione che realizzeremo, potremo attivare i sensori a livelli di luce diversi. I livelli si impostano girando questi potenziometri.

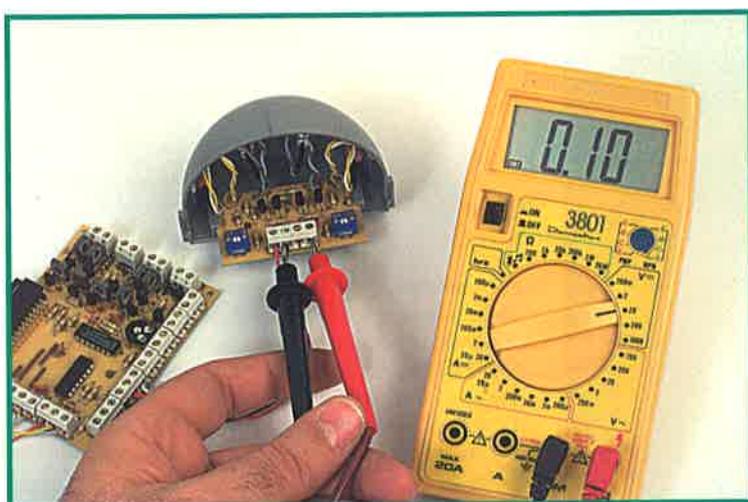
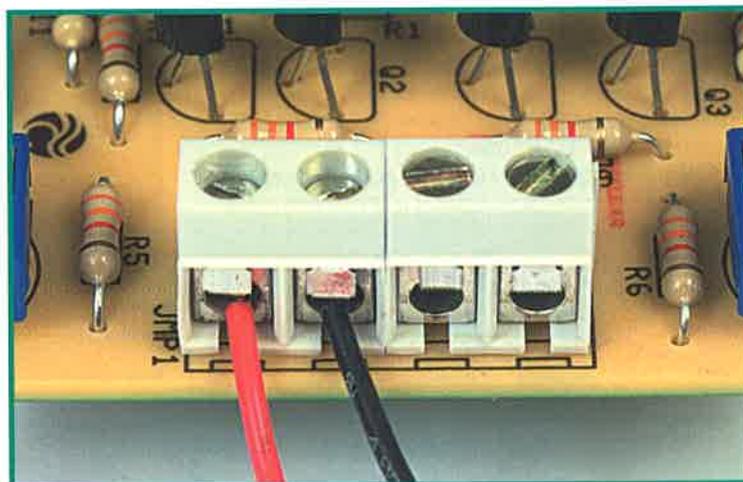


Per realizzare le regolazioni utilizzeremo un cacciavite con cui muovere facilmente il cursore di ogni potenziometro in entrambe le direzioni. Potremo fare ombra con la mano per trovare qual è il punto in cui il diodo LED si illumina e fissare il punto in cui vogliamo che lo faccia. Se il LED si illumina significa che il sensore si è attivato, e che un segnale logico TTL arriva al microprocessore.



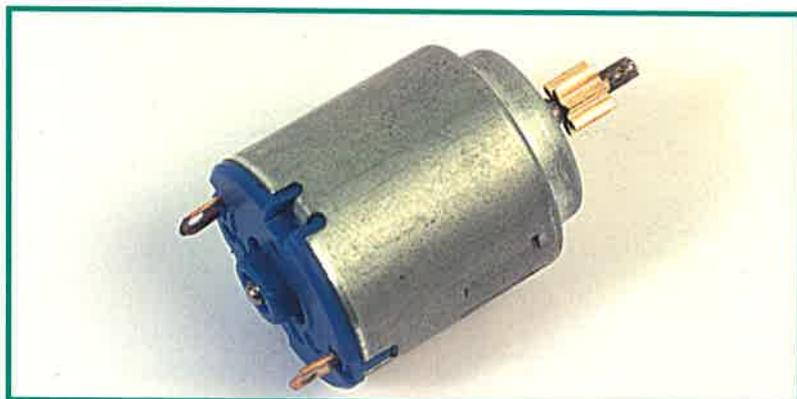
Questo è lo schema elettrico che vale per entrambi i sensori di cui è dotata la scheda delle luci. Questo schema fa sì che il sensore funzioni digitalmente, in modo che produca uno zero logico alla sua uscita raggiungendo un determinato livello di luce. In funzione della quantità di luce che arriva a D3 e del valore di R13, si applica una tensione su Q1, che quando sarà sufficiente, polarizzerà il transistor e attiverà il circuito digitalizzatore formato da Q2. Si tratta in realtà di un circuito a scatto o "Schmitt-Trigger".

La scheda delle luci si collega tramite la scheda di potenza, allo stesso modo della scheda dei sensori. È necessario fornire i 5 V di alimentazione e la massa. Come si può vedere nell'immagine, nella prima posizione della morsettiera J1 iniziando da sinistra collegheremo i 5 V, e nel secondo la massa. Negli altri due morsetti abbiamo i segnali TTL di uscita corrispondenti ai due sensori di luce.



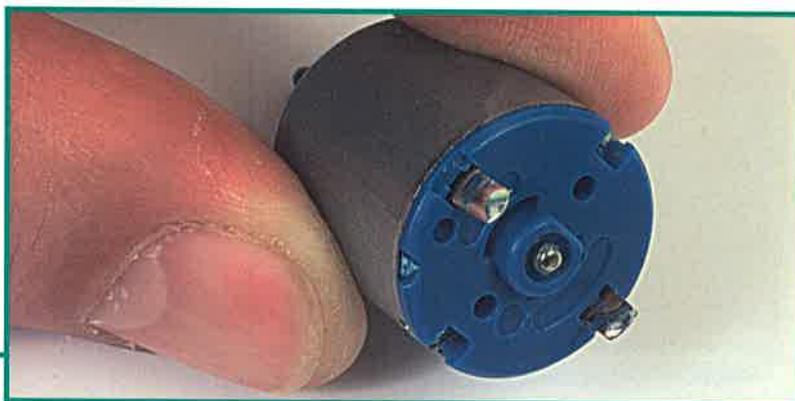
Per verificare il corretto funzionamento della scheda, la alimenteremo tramite la scheda di potenza. A questo punto collegheremo il tester, regolato per misurare la tensione continua sulla scala 20 V, fra uno dei due morsetti di uscita e la massa. Ogni volta che si attiva il sensore, cioè quando si illumina il LED, dovremo vedere sul morsetto il corrispondente segnale TTL di uscita. Questo segnale sarà circa 0 V o +5 V a seconda che il sensore riceva più o meno luce. Queste uscite si possono collegare con gli ingressi della scheda di controllo che sono disponibili (come J6, J7, J8 o J9), secondo l'applicazione che stiamo realizzando.

Montaggio passo a passo



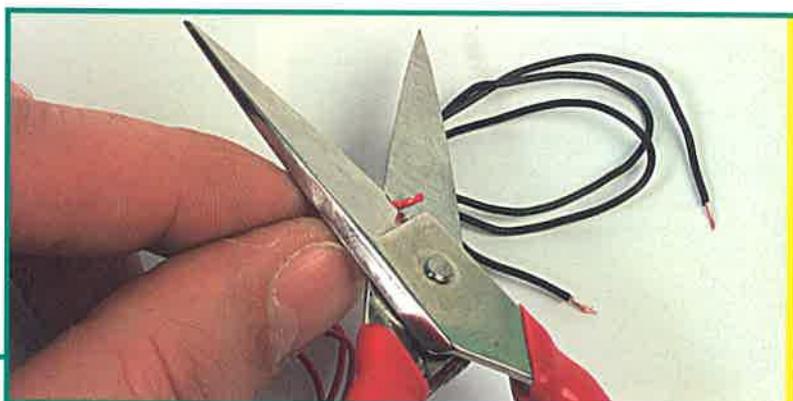
Questo è il secondo motore di cui è dotato Monty, per il suo sistema di trazione e di direzione di movimento. Dispone di un asse provvisto di un pignone, preparato per essere attaccato al sistema di trazione.

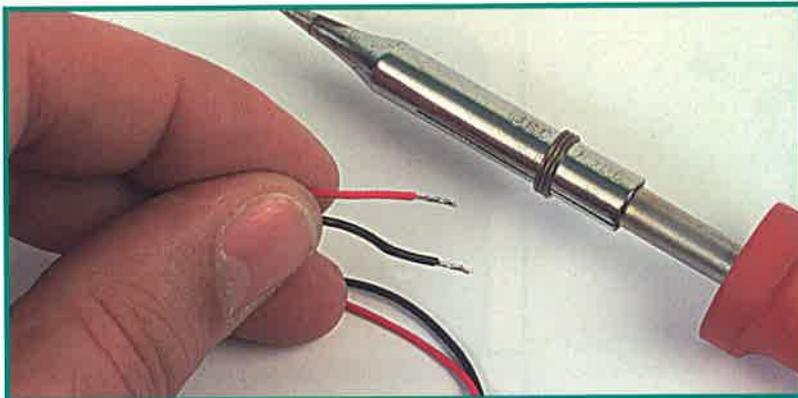
Questa è la parte posteriore del motore. Tramite i due terminali, o linguette, si applica una tensione che farà sì che il motore giri in un verso o nell'altro, a seconda della polarità applicata. Queste linguette devono essere prestagnate, per poterle successivamente saldare con i relativi cavi.



Disponiamo di due cavi da 30 cm, con cui realizzeremo i collegamenti fra il motore e la scheda di potenza. Uno è di colore rosso e l'altro nero.

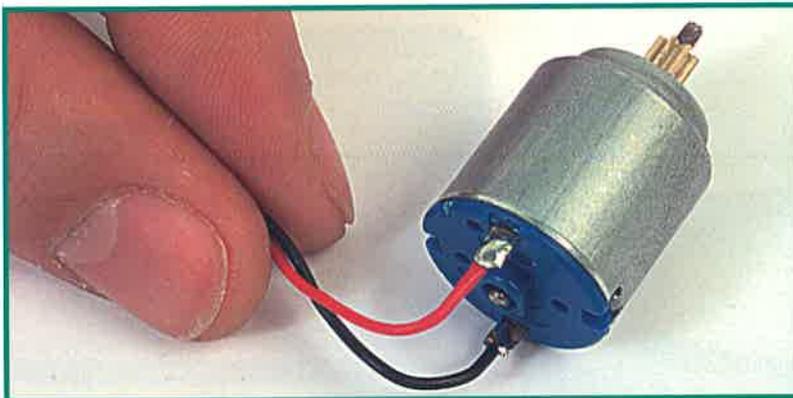
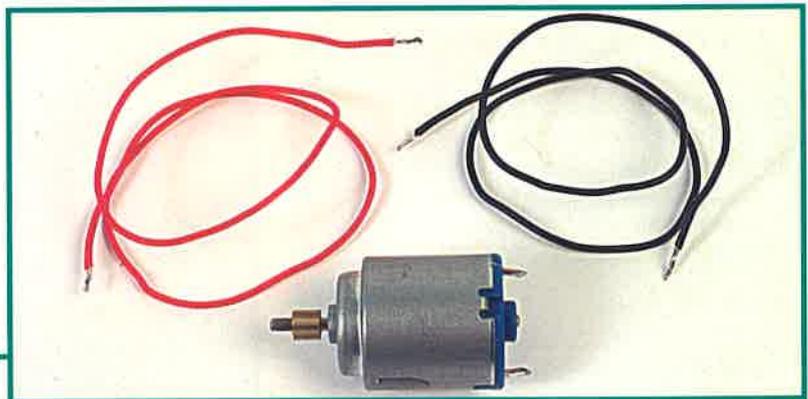
Speleremo, mediante un paio di forbici o di uno spelafili, gli estremi dei due cavi. Raccomandiamo di non fare troppa pressione sulla parte isolante dei cavi, per non danneggiare il rame conduttore che si trova all'interno.





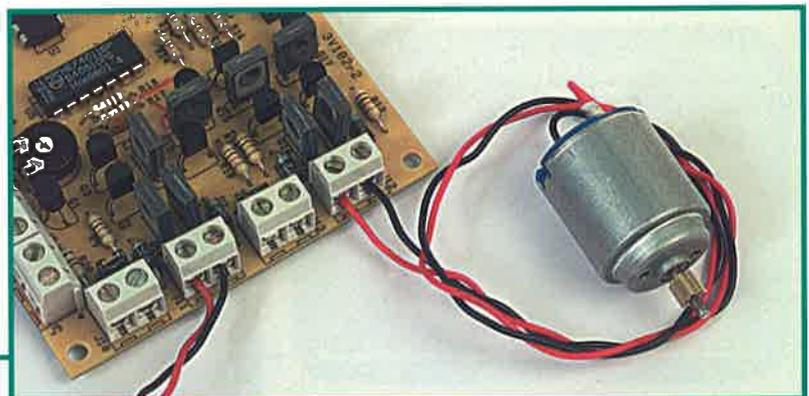
Gli estremi che vanno saldati alle linguette del motore si spelano di circa 3 mm. Se si spelano troppo potrebbe capitare che, in caso di torsione, si tocchino fra di loro i conduttori, provocando un cortocircuito. In seguito si intrecceranno e prestagneranno.

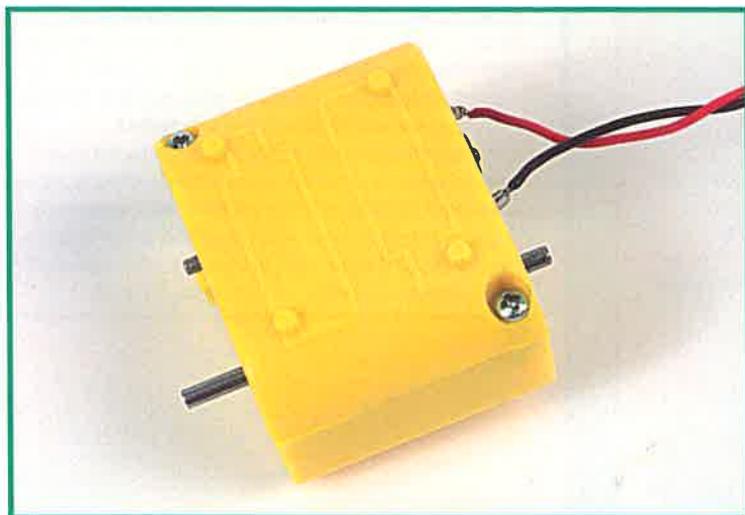
Gli altri estremi dei cavi si collegheranno con la scheda di potenza. Da questo lato i cavi si speleranno di circa 5 mm, si intrecceranno e si prestagneranno. L'immagine mostra i due cavi pronti per essere utilizzati.



Se osserviamo le linguette, vedremo che una delle due è vicina a un segno a forma di punto, o di cerchio. In essa salderemo il cavo rosso, e nell'altra quello nero. A questo punto il motore è già pronto per essere introdotto nel telaio e collegato alla scheda di potenza.

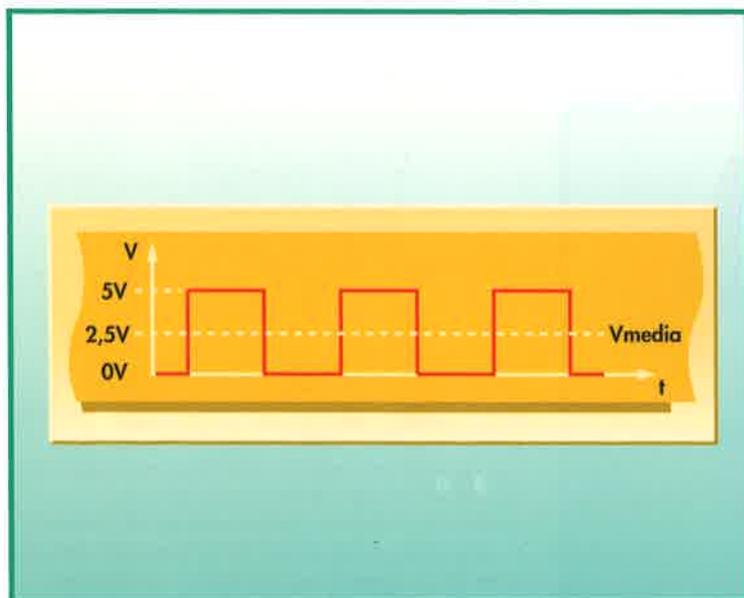
Gli altri due estremi dei cavi del motore si collegheranno alla scheda di potenza. Questa connessione si può realizzare sulle morsettiere J11 o J12, in modo che funzioni come motore sinistro o destro. Tutto questo dipenderà da come posizioneremo il motore alla fine, sul telaio di Monty. Nell'immagine, a titolo di esempio, è stato collegato a J12 (motore destro).





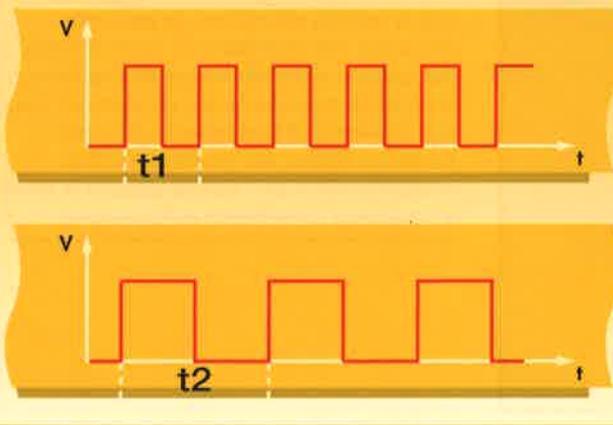
La velocità di rotazione di un motore a corrente continua è funzione della tensione di alimentazione che applichiamo ai suoi capi. Tanto maggiore sarà la tensione applicata, quanto maggiore sarà la velocità del motore. Se parliamo dei motori che controllano le ruote del robot, quanta più tensione applicheremo ad esse (all'interno di un limite massimo fissato dal motore), maggiore sarà la velocità di spostamento del robot. La velocità inoltre può essere controllata meccanicamente se il motore dispone di un riduttore a ingranaggi come quello dell'immagine.

Ci sono diverse tecniche per ottenere questi cambi di velocità, la più semplice consiste nell'applicare diversi livelli di tensione esterna con pile diverse, o batterie, come quelle mostrate nella figura. Questa è una soluzione hardware.



Potremo anche impiegare soluzioni software per modificare la velocità di rotazione del motore. È una proprietà molto interessante, con cui potremo modificare dinamicamente la velocità del motore lungo il programma, mantenendolo sempre alimentato con lo stesso pacchetto di batterie. Per ottenere questo sfrutteremo l'effetto della tensione media risultante da un treno di impulsi. Con un segnale che varia tra 5 e 0 V, come quello mostrato nell'immagine, il valore della tensione media finale sarà di 2,5 V.

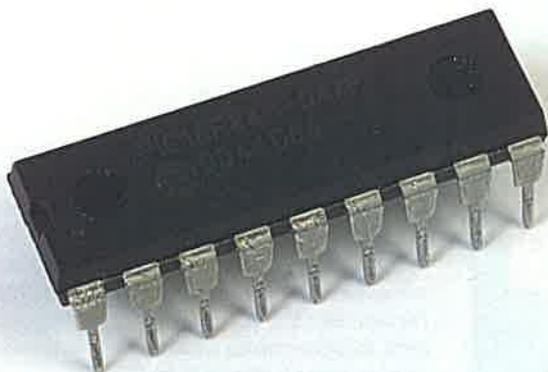
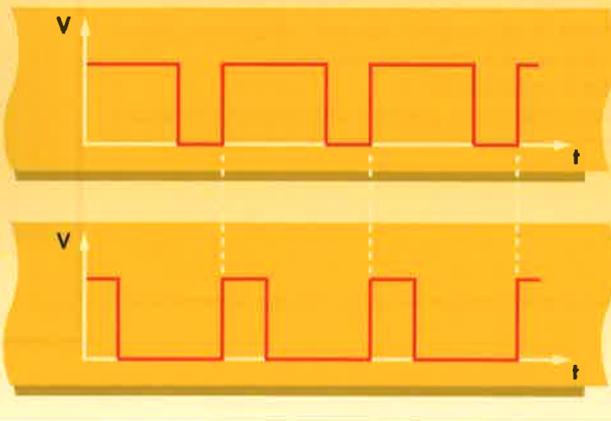
Modulazione di Frequenza



Esiste un'altra tecnica denominata regolazione di tensione mediante la modulazione dell'ampiezza degli impulsi. Consiste nell'utilizzare differenti treni di impulsi per ottenere i valori medi di tensione che possono essere applicati a un motore, il microcontroller sarà incaricato di generare questi treni di impulsi, e pertanto sarà quello che controllerà la velocità di rotazione del motore. Nell'immagine sono mostrate diverse modulazioni di ampiezza degli impulsi basate sulla frequenza, cioè, variando il tempo fra un'onda e la successiva (t_1 e t_2).

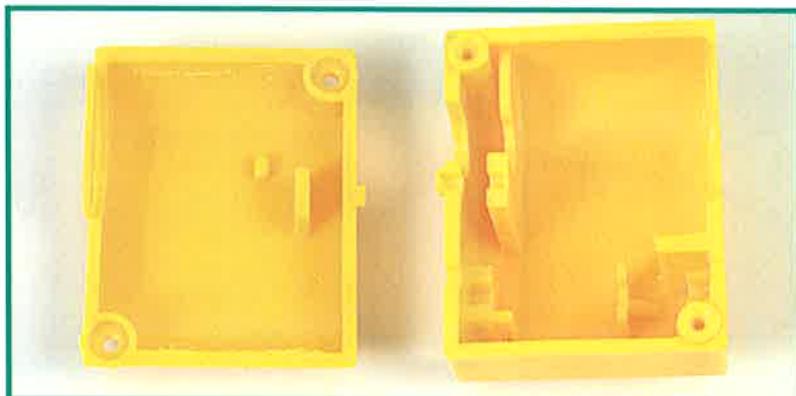
Oltre alla variazione di frequenza, un'altra tecnica molto comune consiste nel modificare quello che viene chiamato ciclo di lavoro. Si utilizza una forma d'onda a frequenza costante, e si modifica solo la parte positiva dell'onda. Tanto più dura il semiciclo positivo, tanto più alta sarà la tensione media finale che si applica al motore.

Modifica del Ciclo di Lavoro



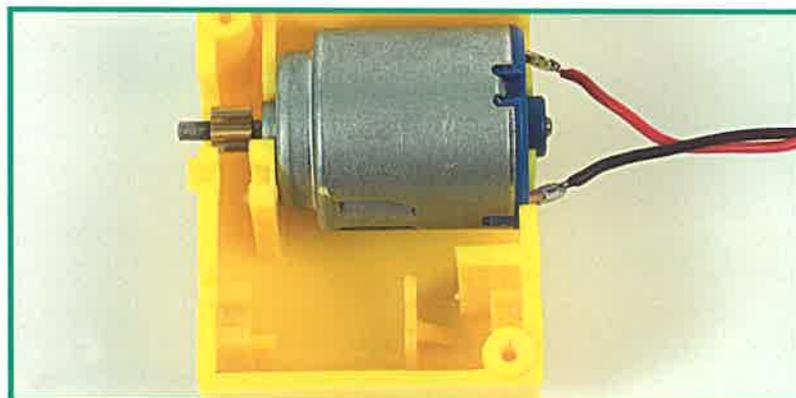
Un microcontroller come quello dell'immagine è un dispositivo intelligente. Grazie ad esso potremo generare diverse forme d'onda, i cui valori medi ci serviranno per modificare la tensione da applicare al motore e quindi modificarne la velocità. Questa caratteristica è molto interessante, perché potremo fare sì che la velocità del robot sia più bassa nelle curve, e raggiunga il massimo nei rettilinei.

Montaggio passo a passo



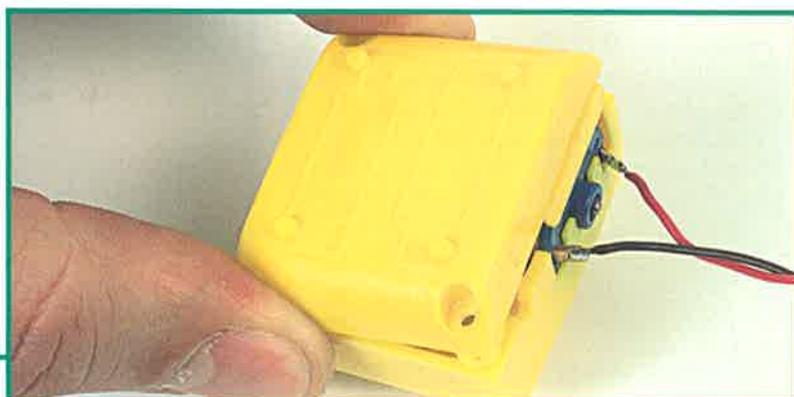
Questa è una delle due scatole con cui chiuderemo il motore, e gli alberi con gli ingranaggi che formeranno il suo sistema di trasmissione. Si compone di due pezzi, come si vede nell'immagine.

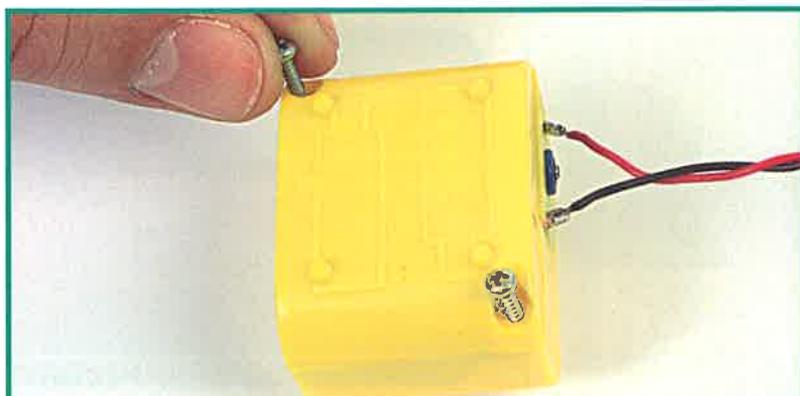
Per chiudere le due parti della scatola del motore e fissarne la struttura, dovremo utilizzare due piccole viti. Avremo perciò bisogno di un cacciavite con punta a stella.



Questa è la posizione in cui va inserito il motore all'interno della scatola. Si incastra solo se inserito nel verso mostrato nella figura in una delle due metà.

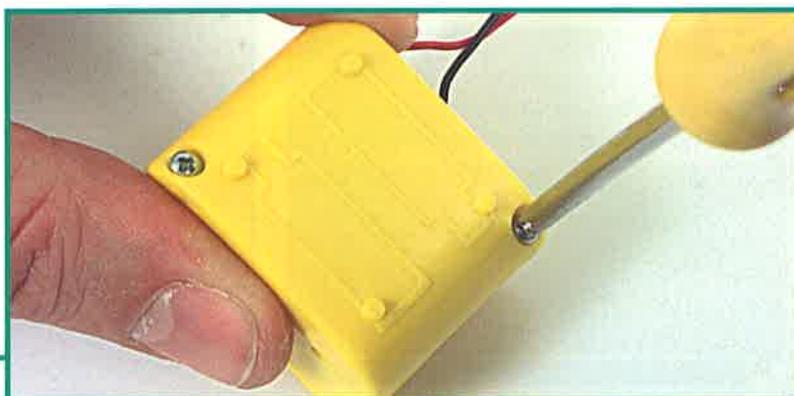
Una volta introdotto il motore potremo chiudere la scatola. Per questo collocheremo l'altra parte della scatola sul motore. Questo secondo pezzo di scatola ha anch'esso un'unica posizione di inserzione. Faremo questa operazione più avanti, una volta che avremo montato gli ingranaggi del riduttore.



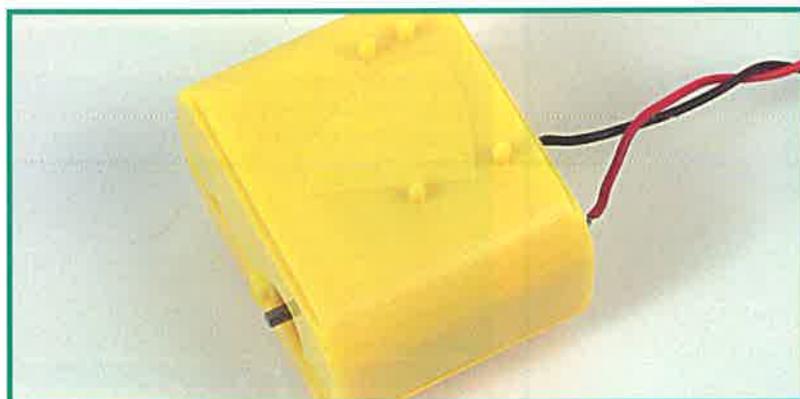


Una volta posizionata la scatola e aver chiuso completamente la struttura, introdurremo le due viti sulla parte superiore del telaio. Questo andrà fatto più avanti, quando collocheremo gli ingranaggi del riduttore.

Chiuderemo le viti con un cacciavite a stella fino a che siano ben fissate e possano sostenere in modo fermo la struttura del telaio. Questa operazione verrà effettuata successivamente, quando avremo montato gli ingranaggi del riduttore.



Questo è l'aspetto finale che presenterà il motore una volta inserito nella scatola. Resterà chiuso e pronto per essere utilizzato. La scatola del motore è così pronta per essere introdotta nel telaio, incastrandosi perfettamente nella struttura del robot.



Questa è una ruota del robot. Non dispone ancora della gomma di caucciù esterna che la ricoprirà e servirà da pneumatico. È una delle due ruote che saranno utilizzate per controllare la direzione di movimento del robot.



```
c:\progra~1\mplab\masc_51\eserc.asm
1
2      list    p=16F84
3      RADIX  HEX
4
5      include "P16F84a.inc"
6
7      TEMP   EQU    0X0E
8      VELOCITA` EQU    0X0F
9
10
11     ORG    0X00
12     goto  INIZIO
13
14     ORG    5
15
```

Realizzeremo un esercizio pratico in cui modificheremo la velocità di rotazione del motore mediante la modulazione dell'ampiezza degli impulsi; utilizzeremo il microcontroller per generare il treno di impulsi. Come tutti i programmi, inizieremo definendo il tipo di dispositivo e le etichette che vogliamo impiegare. Definiamo anche il vector di Reset.

Questa è la routine di ritardo che utilizzeremo nel nostro programma, per generare il treno di impulsi e modificare il valore della tensione media che si applica al motore. Realizziamo una modulazione di ampiezza di impulsi, che modificherà la frequenza del treno, per generare distinti valori medi di tensione.

```
c:\progra~1\mplab\masc_51\eserc.asm
15
16 ;*****
17 ;Routine di RITARDO.
18
19 RITARDO10      bcf    INTCON, 2
20                movlw  0XD8
21                movwf  OPTION_REG
22
23 RITARDO_10     btfss  INTCON, 2
24                goto   RITARDO_10
25                decfsz TEMP, 1
26                goto   RITARDO10
27                return
28
29 ;*****
```

```
c:\progra~1\mplab\masc_51\eserc.asm
29 ;*****
30
31 INIZIO        bsf    STATUS, RP0
32                clrfs PORTA
33                clrfs PORTB
34                movlw b'00000010'
35                movwf OPTION_REG
36
37                bcf    STATUS, RP0
38
39 ;*****
```

La prima cosa che faremo nel programma sono le configurazioni. In questo caso configureremo la porta A come uscita, dato che dobbiamo gestire il motore M1 collegato a J11 della scheda di potenza. Assegniamo un prescaler al Timer0, operazione che si realizzerà nel registro OPTION del banco 1.

```

c:\progra~1\mplab\asc_51\eserc.asm
38
39 ;*****
40
41          movlw    .02
42          movwf    VELOCITÀ
43
44 ;*****
45

```

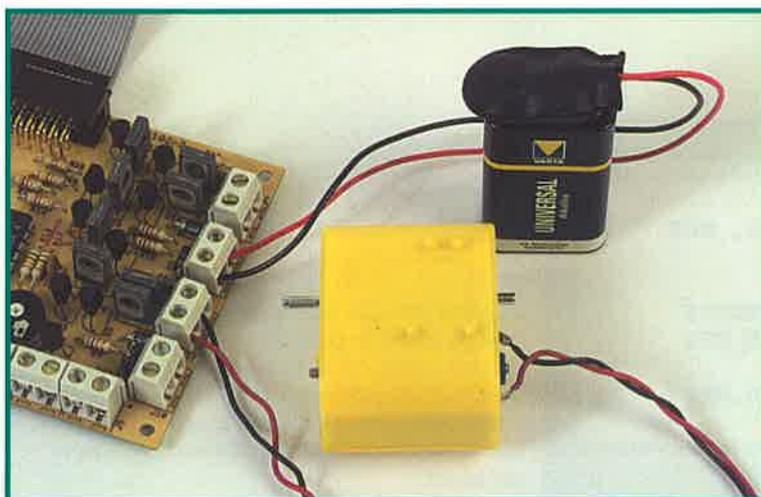
A seconda del valore che introdurremo nella variabile VELOCITÀ, faremo sì che il motore si muova a maggiore o minore velocità. Tanto maggiore sarà il valore di questa variabile, più rapido girerà il motore. Dobbiamo ripetere l'esercizio modificando i valori di questa variabile, osservando e comprendendo quale sia il comportamento del motore in ogni caso.

Questo è il ciclo principale del programma. Consiste nell'accendere il motore, realizzare la temporizzazione corrispondente, funzione della variabile VELOCITÀ, e in seguito tornare a spegnere il motore, temporizzare e ripetere nuovamente il ciclo.

```

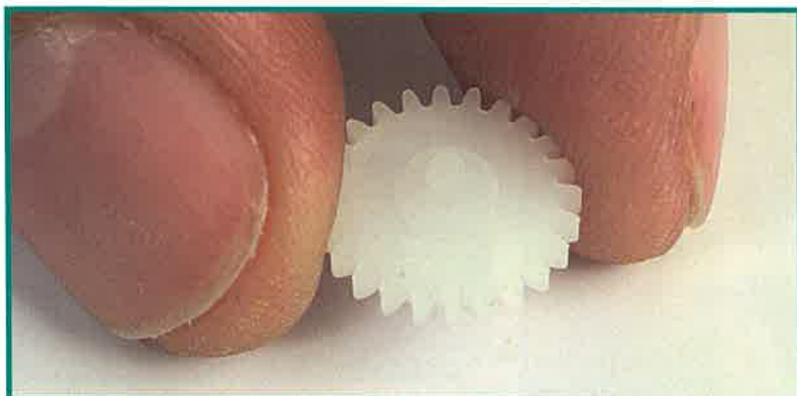
c:\progra~1\mplab\asc_51\eserc.asm
45
46 VEL          movf    VELOCITÀ, W
47              movwf   TEMP
48              movlw   b'00000001'
49              movwf   PORTA
50              call    RITARDO10
51
52              movf    VELOCITÀ, W
53              movwf   TEMP
54              clrf    PORTA
55              call    RITARDO10
56
57              goto    VEL
58
59              END
60

```



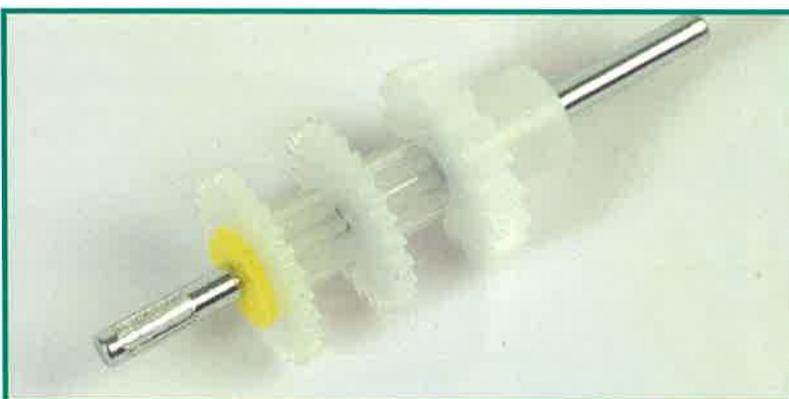
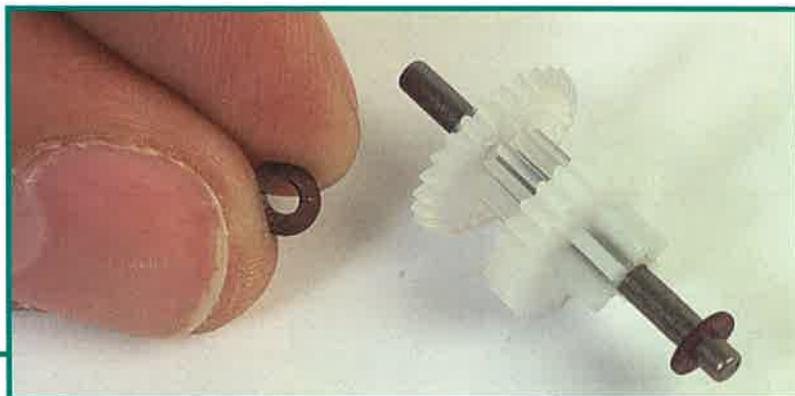
La connessione del motore si realizza tramite la scheda di potenza, come abbiamo fatto con gli altri esercizi. A sua volta la scheda di potenza deve essere collegata alla scheda di controllo, dove si trova il chip programmato. Il lettore può fare pratica effettuando un controllo della velocità di rotazione del motore mediante modifica del ciclo di lavoro del treno di impulsi, invece di modificare la frequenza.

Montaggio passo a passo



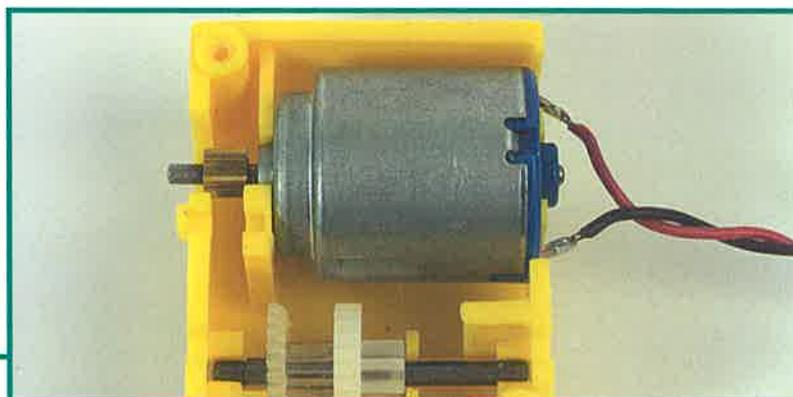
Nell'immagine si vede uno dei tre ingranaggi che formano parte del sistema di riduzione dei motori. Si tratta di un ingranaggio molto piccolo. Questo si incastra direttamente con il pignone del motore e trasmette il movimento al resto dell'asse.

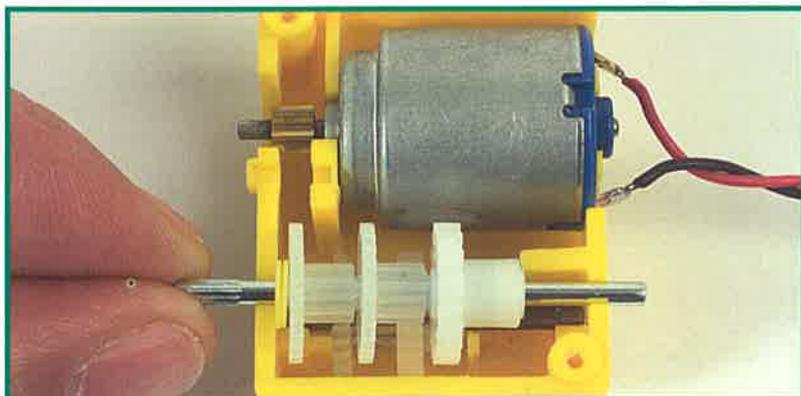
L'immagine mostra l'asse centrale con i suoi due ingranaggi. È possibile che i due estremi degli assi dispongano di una rondella di protezione per evitare che escano gli ingranaggi dallo stesso. Questa rondella dovrà essere tolta al momento di collocare l'asse all'interno del telaio. Inoltre è conveniente applicare del lubrificante, o del grasso, per evitare attriti e ottenere un maggior rendimento nel sistema di riduzione del motore.



Questo è l'asse più lungo dei tre, e ha il compito di trasmettere il movimento del motore al sistema di trazione o ruote di Monty. Dispone di tre ingranaggi, come si vede nell'immagine. In uno degli estremi dell'asse dispone di una rondella, che non toglieremo, per evitare attriti fra l'ingranaggio e il supporto. Inoltre è conveniente applicare del lubrificante o del grasso.

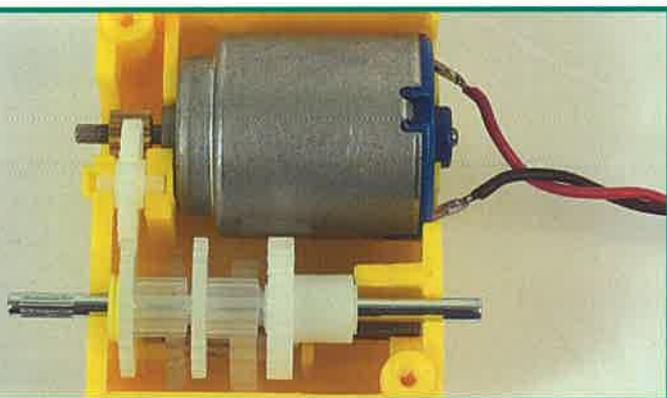
Come si vede nell'immagine, nella scatola si colloca per prima cosa l'asse centrale, che deve essere fissato in modo sicuro ai supporti corrispondenti. A questo punto, possiamo fissare il motore nell'apposita sede.





L'asse più grande, che dispone di tre ingranaggi, si colloca sui supporti corrispondenti. Gli ingranaggi di questo asse devono accoppiarsi con gli ingranaggi dell'asse centrale, che è stato collocato in precedenza. Nell'immagine si può vedere il verso in cui è stato inserito questo asse, perché si incastrerà in modo adeguato con i pignoni.

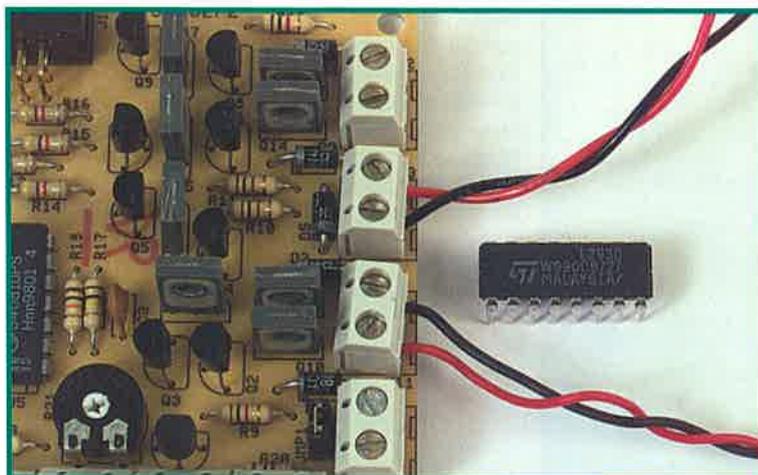
Come si vede nell'immagine, l'asse più piccolo dispone di un unico ingranaggio. Questo ingranaggio deve essere fissato all'asse e perfettamente centrato al medesimo.



L'asse più piccolo sarà l'ultimo che monteremo, e deve rimanere nella posizione mostrata nella figura. Questo asse accoppia il movimento del pignone del motore con il resto del sistema degli ingranaggi, producendo un effetto di riduzione di velocità sull'asse più lungo, sul quale si fissa la ruota di Monty. Ora verificheremo che il sistema di ingranaggi giri correttamente e chiuderemo la scatola, come spiegato in precedenza. Il sistema motore/riduttore è ultimato.

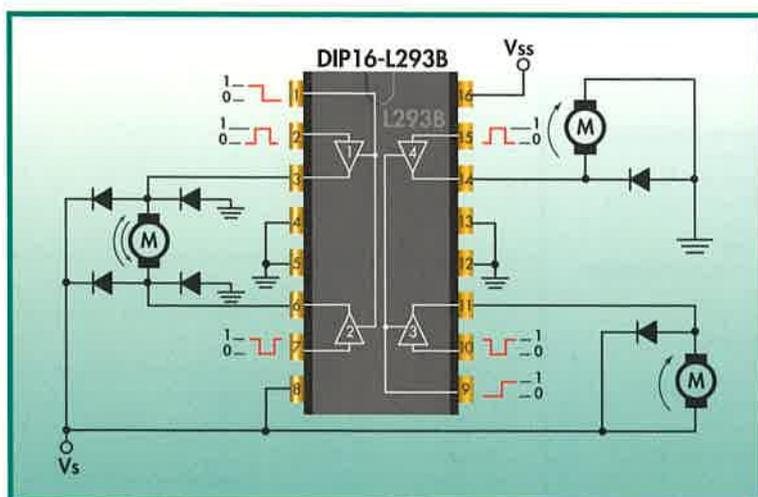
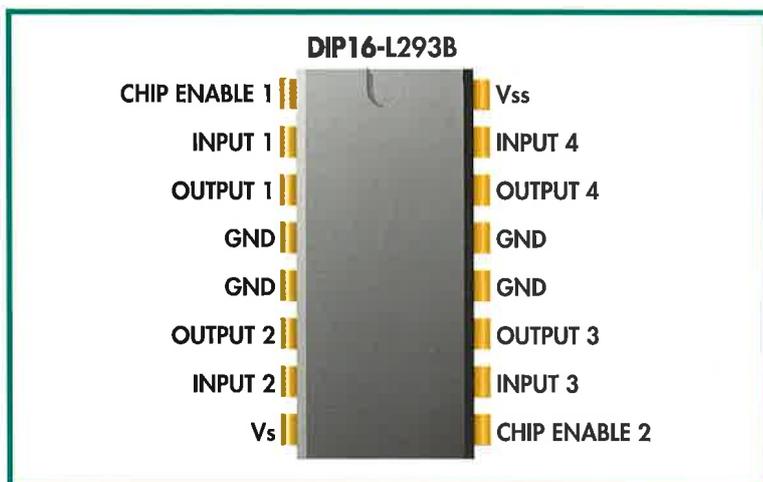
Questi sono il coperchio delle pile e il coperchio del ruotino posteriore. Serviranno per supportare entrambi gli elementi sul telaio del robot. Si collocano nella parte inferiore del telaio stesso.



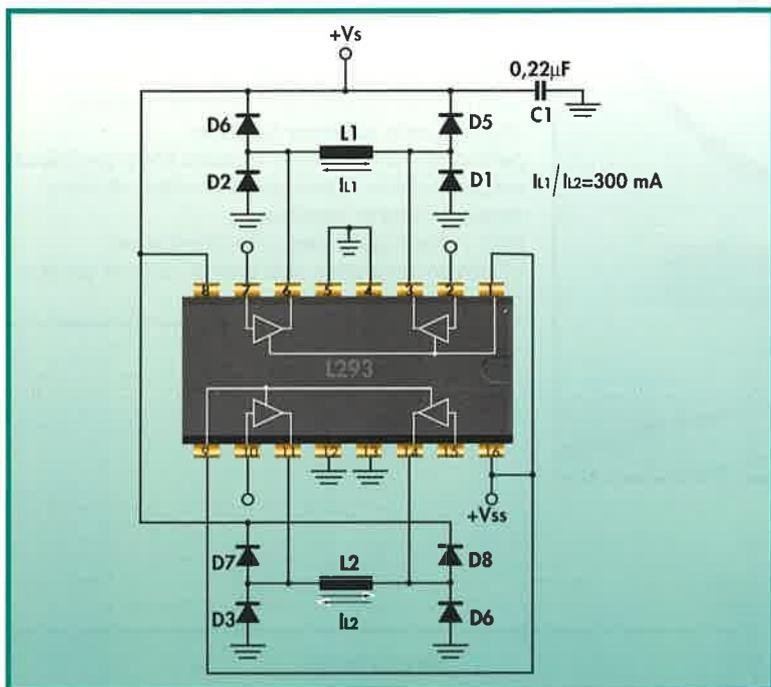


La scheda di controllo di Monty è costituita da un ponte ad H di transistor che permettono di realizzare il controllo dei motori, tanto per l'accensione e lo spegnimento, quanto per il senso di rotazione. Oltre a questa soluzione hardware per il controllo dei motori, esistono chips specializzati che permettono di realizzare il controllo di piccoli motori a corrente continua. Sono i drivers per motori, che incorporano all'interno del proprio chip tutte le funzioni dei due ponti ad H di transistor.

Questa è la piedinatura del driver per motori L293B. Questo tipo di chip è consigliato per quei lettori che vorranno potenziare le funzioni di Monty, aggiungendo dei motori in più, dato che con solo due chip se ne possono pilotare di diversi. Il circuito L293b permette inoltre di applicare al motore una tensione di lavoro totalmente indipendente dalla tensione di alimentazione, che si introduce tramite il Pin 8 del chip (V_s).

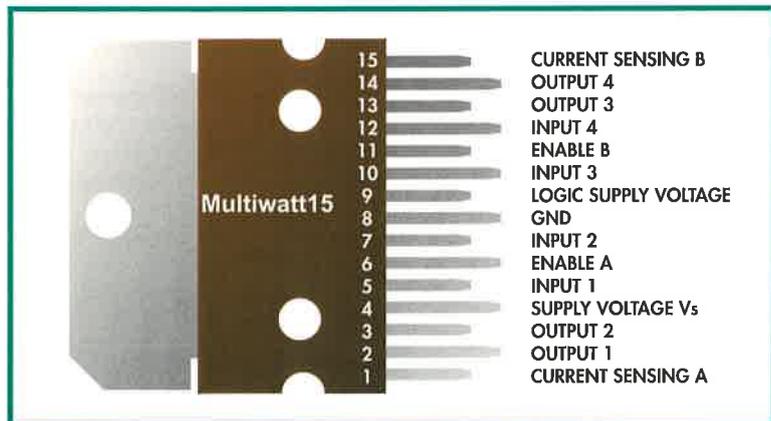
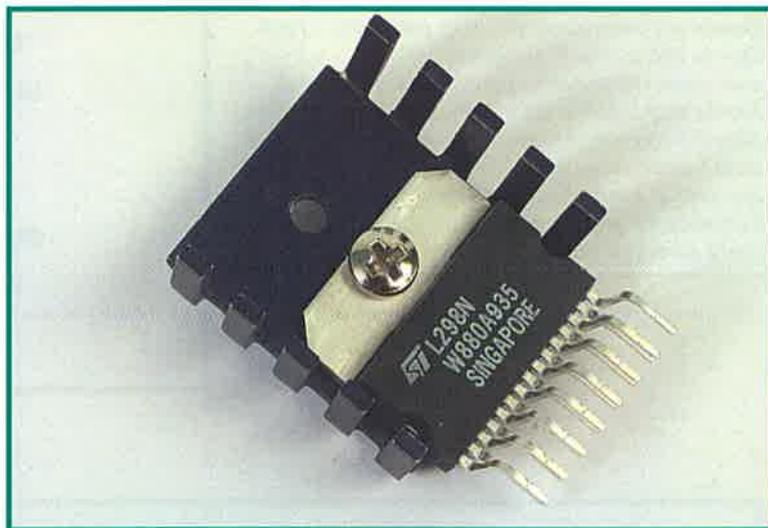


La figura mostra i differenti tipi di collegamenti tra il driver L293B e il motore a corrente continua. Mediante i segnali di Enable (En_x), si abilita o disabilita il driver, controllandone la possibilità di accenderlo e spegnerlo. Gli In_x collegano, o meno, il motore associato ad ogni singolo canale. Un motore può essere associato a due canali. In questo modo si può decidere il verso del giro. Per fare questo i corrispondenti ingressi In_x devono avere un valore complementato (1-0 o 0-1). Tutti questi segnali si governano con livelli TTL direttamente dal microcontroller.



Questi driver sono particolarmente utili per la gestione dei motori passo a passo (PAP). Questi motori hanno più bobine dei motori CC convenzionali, quindi richiedono come minimo quattro ponti ad H di transistor. Invece se utilizziamo un driver, abbiamo in un solo chip tutte le funzionalità necessarie per la gestione di questo tipo di motori. Nella figura si vede il modo di collegare un motore PAP al driver L293B.

Il driver che si vede nell'immagine è l'L298. Questo chip ha il doppio della potenza dell'L293B, sopportando fino a 4 A per canale. È particolarmente raccomandato se vogliamo aggiungere a Monty un altro motore in grado di realizzare una funzione che richieda alti consumi, come sollevare oggetti piuttosto pesanti. In questa situazione il motore richiede un alto consumo ed è quindi necessario utilizzare driver speciali ad alta potenza.



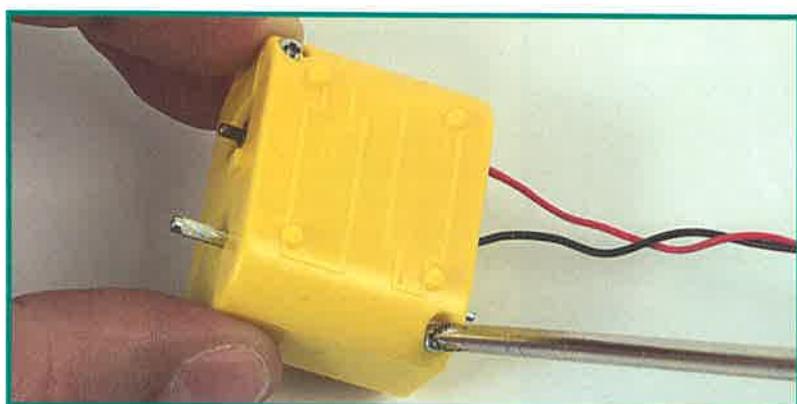
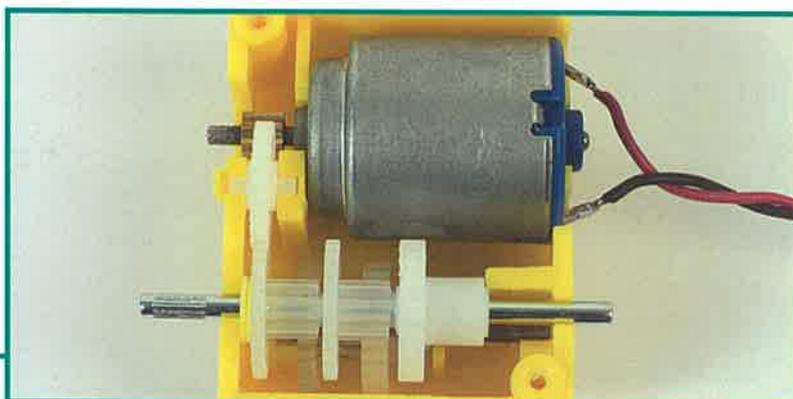
La figura riporta la piedinatura di questo driver. Come possiamo vedere è simile all'L293B. Abbiamo un ingresso per la tensione di alimentazione dei motori, indipendente dalla tensione di alimentazione. Inoltre disponiamo di doppi canali, per due motori CC o un motore PAP; segnali di Enable (Enx) per fermare o attivare il motore; segnali di Inx per controllare il senso di rotazione del medesimo.

Montaggio passo a passo



Questo è il sistema di trasmissione/riduttore con cui prepareremo l'altro motore di Monty. Anche questa volta dovremo identificare ognuno degli assi con i suoi diversi ingranaggi. Procederemo con il montaggio del sistema allo stesso modo di come abbiamo fatto in precedenza.

L'immagine mostra come rimane tutto il sistema di trasmissione/riduzione alloggiato all'interno della scatola. Dobbiamo assicurarci che i pignoni si accoppino fra loro e che il movimento dell'asse del motore si trasferisca all'asse grande, dove si fisserà l'altra ruota di Monty.



Mediante un piccolo cacciavite a stella chiuderemo il telaio utilizzando le due viti corrispondenti. A questo punto abbiamo pronto il sistema di trazione di Monty, formato da due motori con i corrispondenti riduttori.

Le due gomme di caucciù mostrate nell'immagine serviranno da pneumatici per le ruote. Grazie a queste gomme le ruote avranno maggiore aderenza e trazione sulle superfici lisce.





Monteremo le gomme sulle ruote, come è mostrato nell'immagine. Tenderemo un poco la gomma, quanto basta per poterla introdurre e incastrarle sulla ruota. Le gomme devono rimanere perfettamente allineate rispetto alle ruote, evitando così movimenti strani.

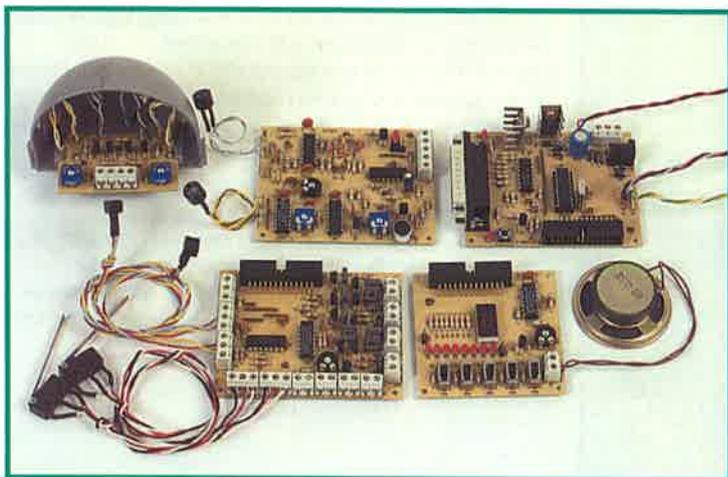
L'immagine mostra la sfera del ruotino posteriore del telaio del robot. Insieme con le ruote anteriori forma i tre punti di appoggio su cui si sostiene il robot. Grazie al suo basso attrito, questo ruotino di forma sferica facilita i diversi giri che può realizzare Monty.



Monteremo il ruotino sul telaio e una volta inserito, monteremo il coperchio del ruotino, per fare in modo che la sfera non esca dalla sua sede e aderisca alla struttura. È un sistema di spostamento simile a quello del mouse del nostro personal computer.

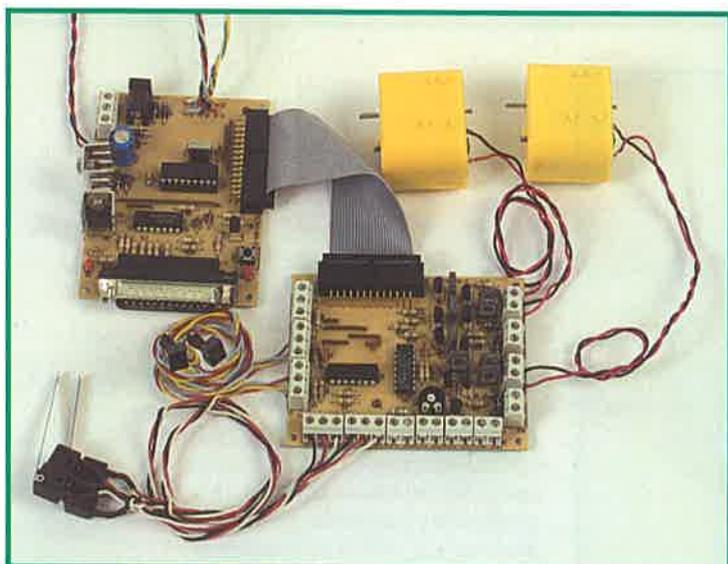
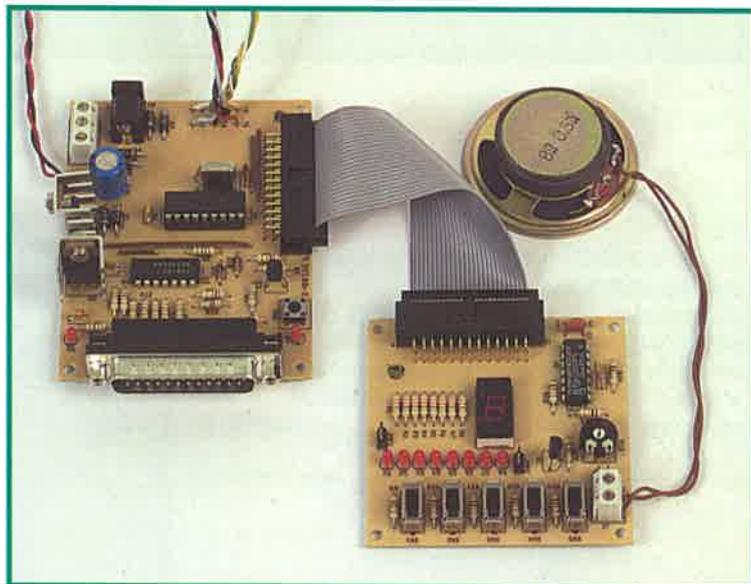
La struttura di Monty è formata da due pezzi, o metà, che formano il corpo, e altre due che costituiscono la testa. Le due metà si fissano fra loro mediante le clips che si vedono nell'immagine, ottenendo in questo modo una struttura chiusa in cui alloggia tutta l'elettronica di controllo. Un pezzo di gommapiuma (anch'esso riportato nell'immagine) permette di ancorare i motori alla struttura producendo l'effetto di "ammortizzatore". Tutto questo vi sarà spiegato successivamente.



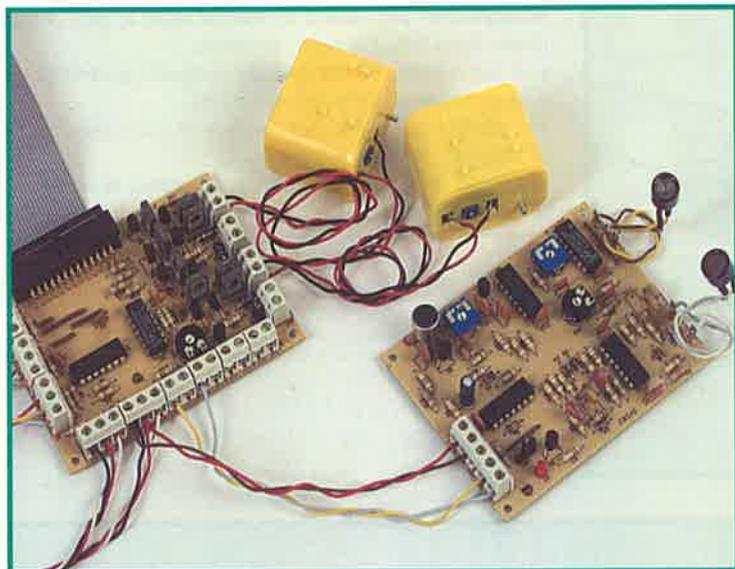


Realizzeremo una visione di insieme delle schede di cui disponiamo fino a questo momento, e di come collegarle fra loro per eseguire gli esercizi e approfittare di tutte le funzionalità del robot. Disponiamo della scheda di controllo, di potenza, di ingressi e uscite, di sensori di luce, e in seguito, costruiremo la scheda di controllo della pinza.

La scheda di controllo è la principale, dato che in essa si trova il microcontroller. Questa è la scheda che si collega con il PC, per realizzare la scrittura del chip, e deve essere sempre presente per qualsiasi esercizio che realizzeremo. La scheda di I/O, anch'essa mostrata nell'immagine, è una scheda di valutazione indipendente per imparare a gestire il microcontroller, però non fa parte delle schede che incorpora Monty.

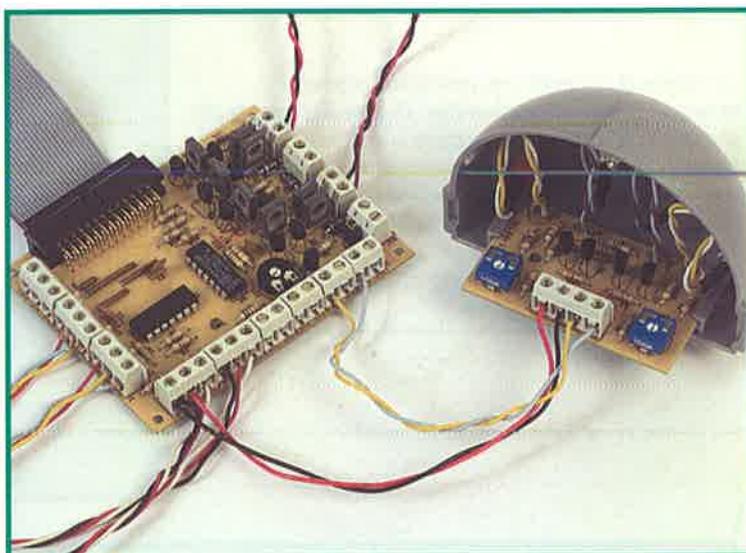


La scheda di potenza è una scheda fondamentale, dato che tramite questa si collegano direttamente i sensori ottici e meccanici del robot, così come i motori di direzione del robot. Questa comunica con la scheda di controllo tramite il cavo PIC-bus. Inoltre, è una scheda indispensabile per qualsiasi esercizio con il robot; tramite questa, collegheremo il resto delle schede.

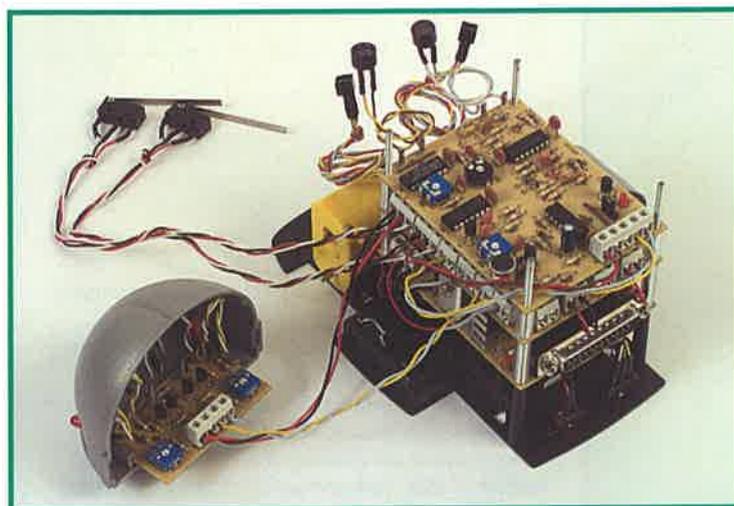


La scheda dei sensori è un optional. Aggiunge al robot le funzioni di ultrasuoni e di rilevazione dei livelli sonori. Si collega tramite la scheda di potenza, dato che non può essere collegata direttamente alla scheda di controllo. Si alimenta tramite i morsetti della scheda di potenza e i segnali si gestiscono tramite le linee di I/O presenti su questi morsetti. Possiamo utilizzare gli ingressi liberi della scheda di potenza (morsetti J6, J7, J8, J9) per realizzare questi collegamenti. Nell'immagine si mostra un esempio di collegamento.

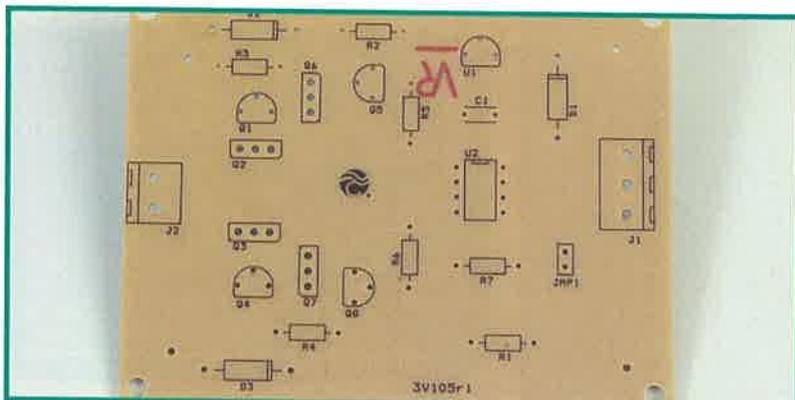
La scheda delle luci, come quella dei sensori e di controllo della pinza, è una scheda che aggiunge funzionalità extra al robot. Così come le altre schede, è sempre collegata al sistema tramite la scheda di potenza, collegando l'alimentazione al segnale di controllo. Si possono utilizzare gli ingressi liberi della scheda di potenza (morsetti J6, J7, J8, J9) per realizzare questi collegamenti.



Nell'immagine sono mostrate le schede di cui disponiamo fino a questo momento, montate, per la gestione del robot. Si può vedere come i motori e i sensori ottici e meccanici si collegano alla scheda di potenza e come le schede dei sensori e delle luci siano collegate insieme, alimentate e controllate tramite le prese dei morsetti liberi nella scheda di potenza.

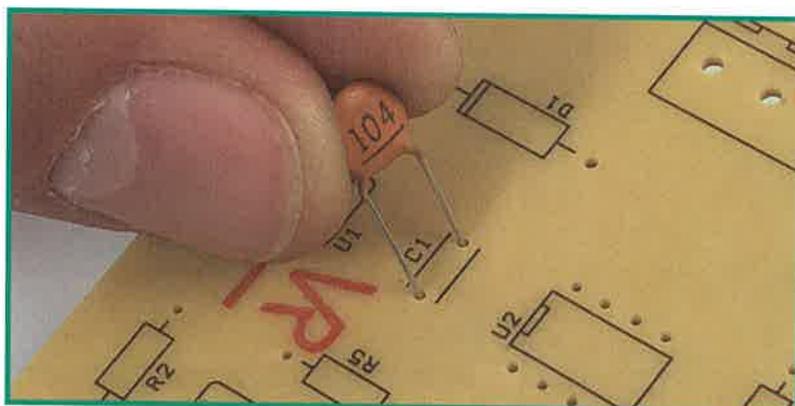
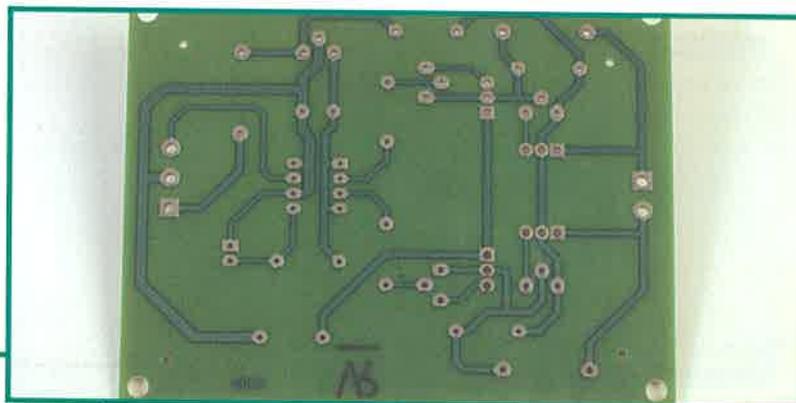


Montaggio passo a passo



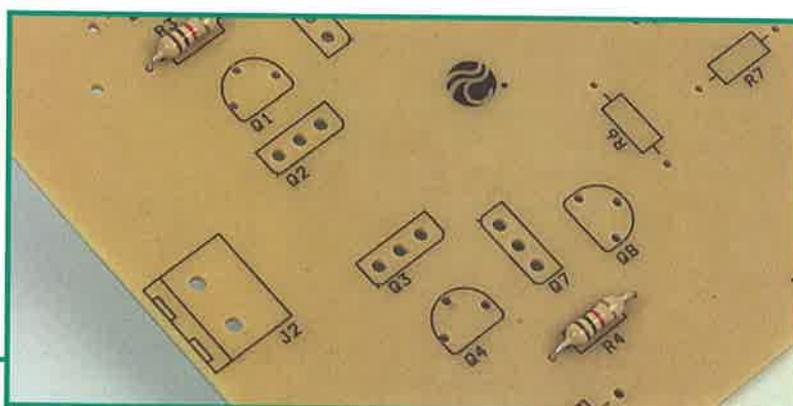
Questa è la scheda di controllo della pinza di Monty. Il robot disporrà di una pinza in uno dei suoi bracci con il quale potrà prendere e trasportare oggetti. Sarà necessario preparare una scheda per il controllo del motore che deve aprire e chiudere la pinza.

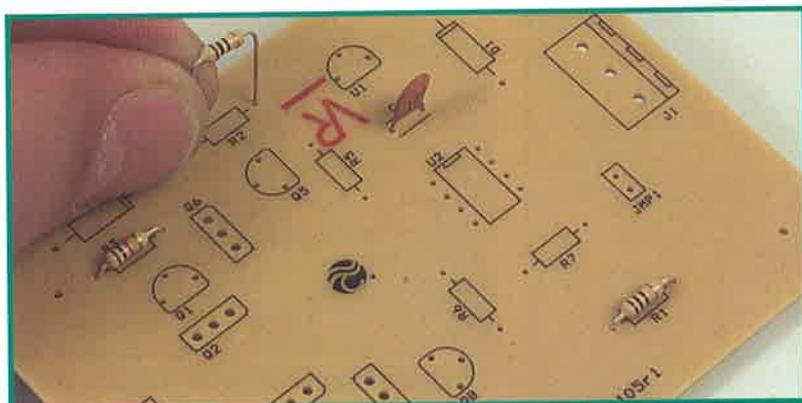
Questa scheda è l'ultima di cui è composto il robot. Ha piste su una sola faccia, come il resto delle schede che abbiamo costruito fino ad ora. Dobbiamo realizzare le saldature con attenzione, secondo il procedimento seguito abitualmente.



Il primo elemento che salderemo è un condensatore da 100 nF. Sarà inserito in C1, ed è l'unico condensatore che ha questa scheda. Non possiede polarità, per cui il suo verso di inserzione è indifferente.

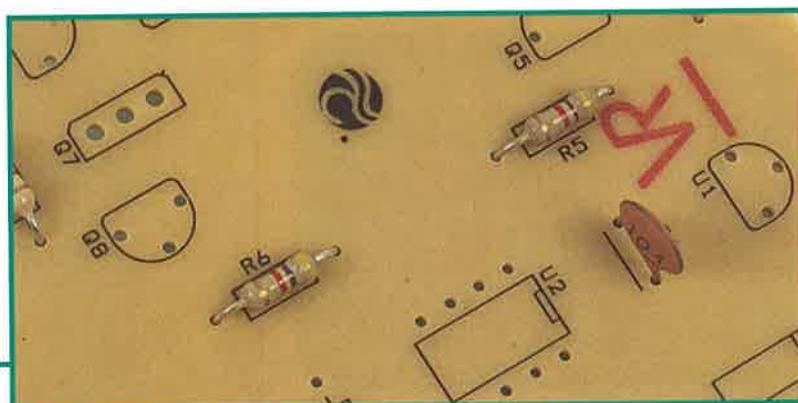
Collocheremo ora due resistenze da 1 k. Le identificheremo tramite il codice colore marrone-nero-rosso. Vanno collocate in R3 e R4.



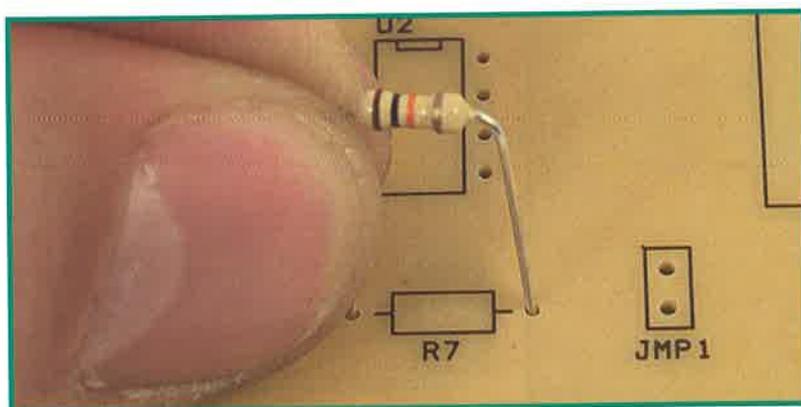


Dobbiamo saldare le altre due resistenze in R1 e R2. Queste resistenze saranno da $10\ \Omega$ e si identificheranno tramite il codice dei colori marrone-nero-nero.

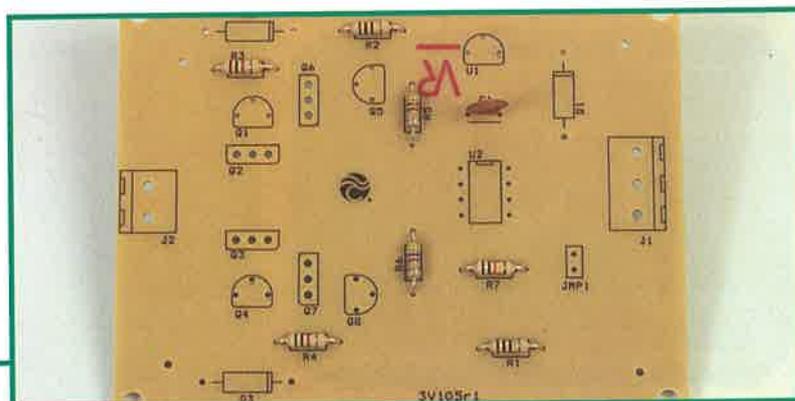
Altre due resistenze da $4,7\ k$ (giallo-viola-rosso) andranno inserite in R5 e R6. Dopo aver saldato le resistenze, taglieremo la parte in eccesso dei reofori.



Infine salderemo l'ultima resistenza di cui è composta questa scheda in R7. Il suo valore è di $10\ k$ e il suo codice colori, già noto, è marrone-nero-arancio.

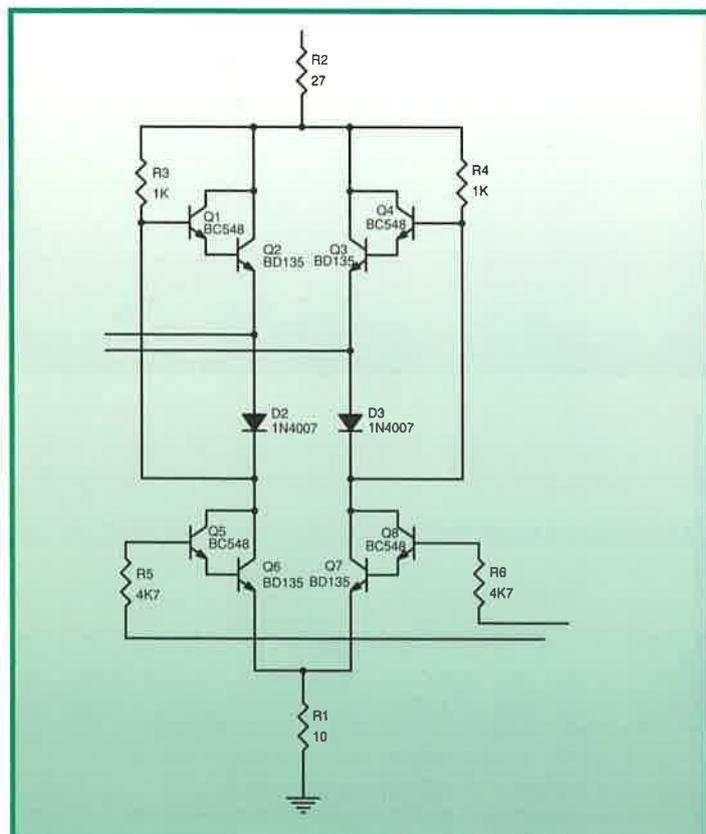
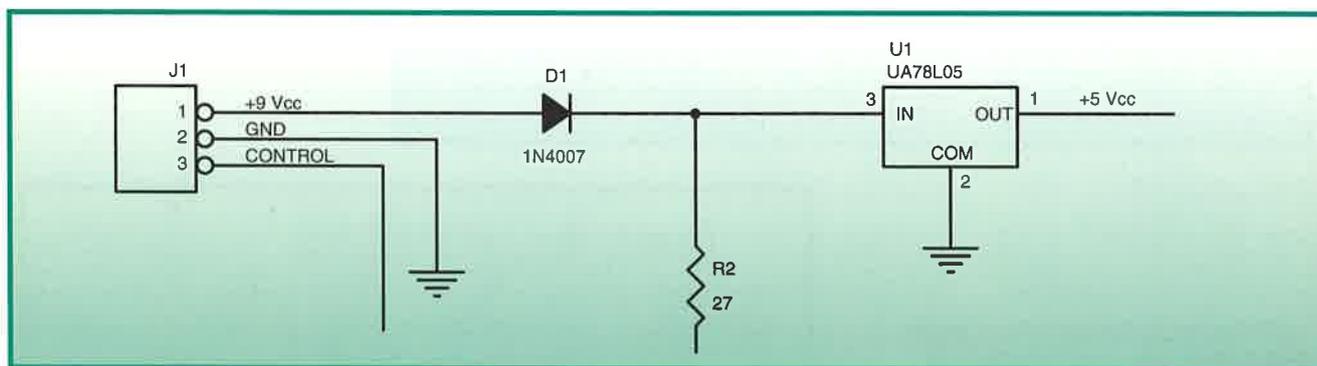


Questa immagine mostra la scheda di controllo della pinza con gli elementi che abbiamo saldato sino a questo momento. Verificate che siano tutti perfettamente ubicati, così come per le saldature, che siano brillanti e a forma di cono.



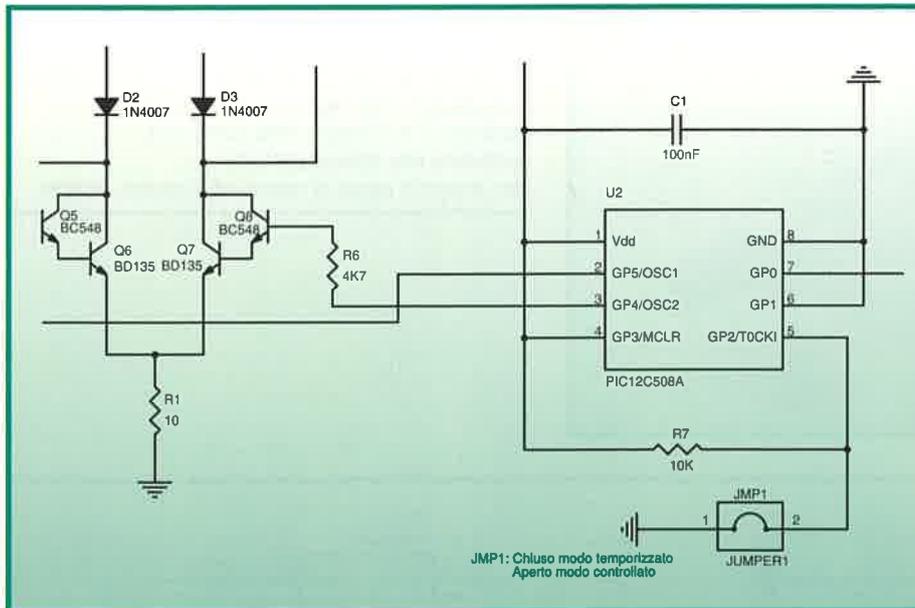


Questa è la pinza di cui sarà dotato il robot in uno dei suoi bracci. Sarà comandata da un motore a corrente continua, e servirà per raccogliere oggetti e depositarli dove vogliamo. Per controllare l'apertura e la chiusura della pinza sarà necessaria una scheda speciale, che governi il senso di rotazione di questo motore.



L'immagine mostra lo schema di collegamento della scheda di controllo alla pinza. Questo si realizza tramite il morsetto J1. Con i piedini 1 e 2 si introduce una tensione di +9 Vcc, con la quale si alimenta il motore. Questa tensione si applica anche al regolatore 78L05 (U1), da cui si ottiene una tensione di +5 Vcc per alimentare il PIC12C508A, incaricato delle funzioni di controllo. Tramite il piedino 3 della stessa morsettiera si applica, dalla scheda di controllo e tramite quella di potenza, il segnale di apertura e di chiusura.

Per la gestione del motore della pinza utilizziamo un ponte ad H, simile a quello della scheda di potenza per il controllo dei motori di direzione del robot. Questo ponte ad H sarà incaricato di amplificare i segnali e alimentare il motore. Tramite questo ponte potremo governare sia l'accensione che lo spegnimento del motore e anche il senso di rotazione con cui la pinza si apre o si chiude.

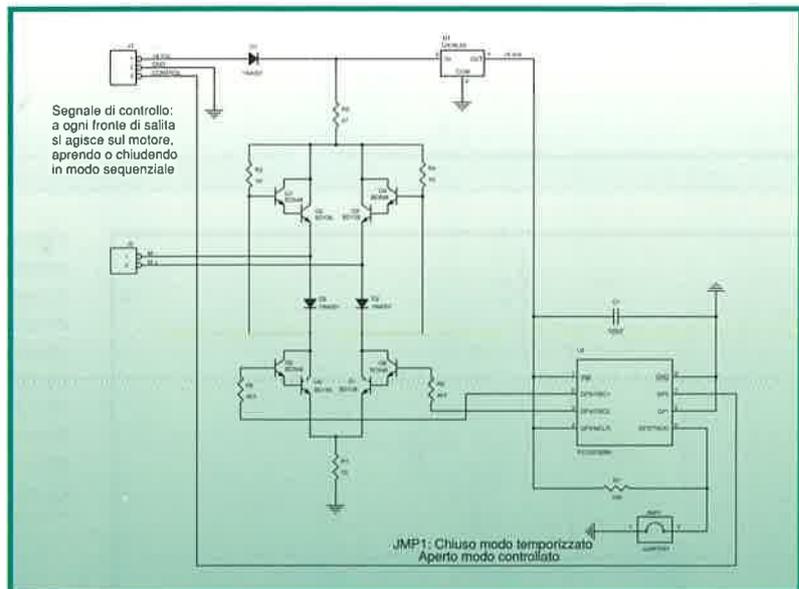


Il ponte ad H che controlla il motore della pinza è pilotato da un PIC12C508, così come si vede nell'immagine.

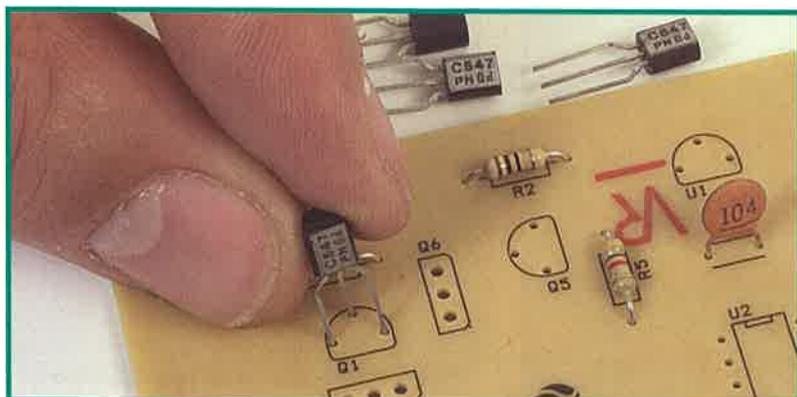
Questo microcontroller viene programmato con programma di controllo. I segnali GP5 e GP4 sono le uscite che terminano al ponte ad H, facendo sì che il motore giri in un verso o nell'altro.

Lo stato del jumper JMP1 determina due forme di funzionamento: in modo controllato (jumper aperto) e in modo temporizzato (jumper chiuso).

Nell'immagine vediamo lo schema globale del controller della pinza del robot. Ogni volta che si collega l'alimentazione, la pinza tende a chiudersi per default. Nel modo controllato, un fronte di salita applicato al pin 3 del morsetto J1 produce l'apertura, un secondo fronte, la chiusura. Nel modo temporizzato l'apertura si produce con fronte di salita e la chiusura è automatica, dopo un tempo di 2" circa. Il motore si collega mediante il morsetto J2.

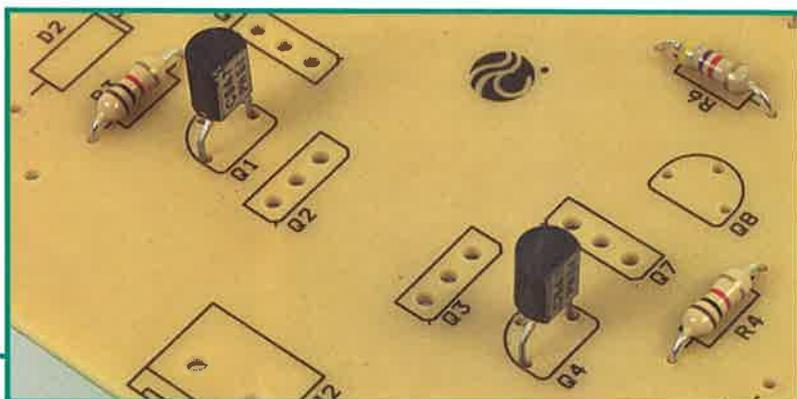


Come si può vedere nell'immagine, la pinza di Monty può sostenere carichi leggeri. Possiamo solo pensare ad applicazioni per cui il robot possa essere utile, e programmarlo perché prenda gli oggetti che vogliamo.

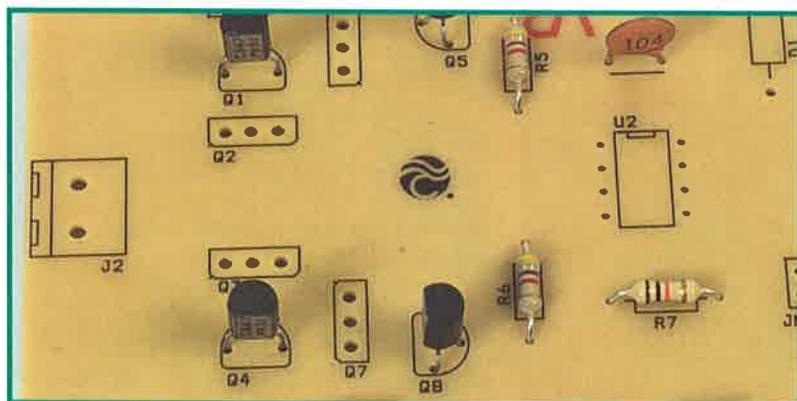


Salderemo 4 transistor modello BC548 sulla scheda della pinza, e monteremo il primo di questi in Q1. È necessario inserire il transistor secondo il disegno a mezza luna rappresentato nella serigrafia.

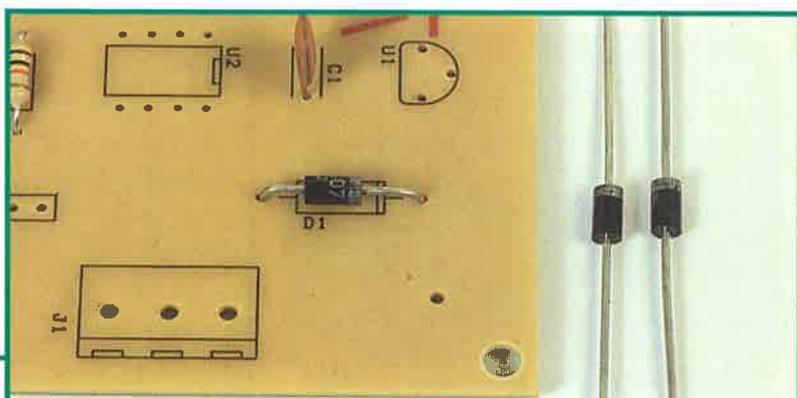
In Q4 inseriremo l'altro transistor modello BC548. I transistor non devono rimanere completamente inseriti nella scheda, ma rialzati di qualche millimetro dalla stessa, come si vede nell'immagine.

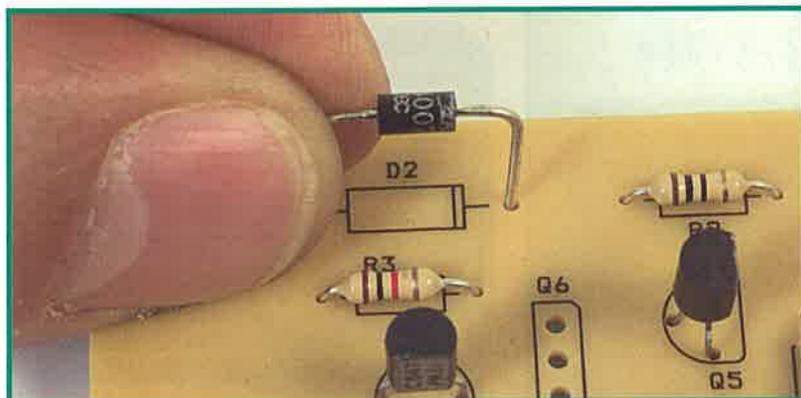


Infine dovremo saldare altri due transistor modello BC548 nelle posizioni Q5 e Q8. Quando avremo saldato i 4 transistor procederemo a tagliare la parte restante dei reofori di ognuno di essi.



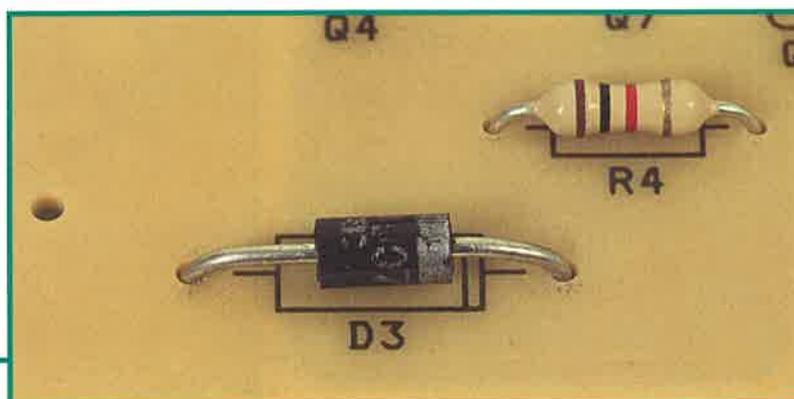
Ora monteremo un diodo modello 1N4007 in D1. I diodi hanno polarità, quindi dobbiamo fare attenzione al verso di inserzione. Devono rimanere come mostrato nell'immagine, facendo coincidere la banda che rappresenta il catodo con quella della serigrafia.



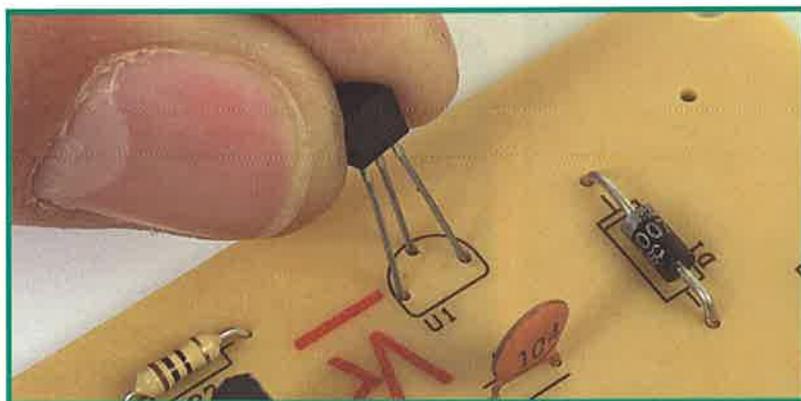


In D2 monteremo un altro diodo 1N4007, inserendolo secondo il verso mostrato nell'immagine, grazie alla sua polarità. Dobbiamo far coincidere la banda grigia del diodo con la linea disegnata nella serigrafia della scheda.

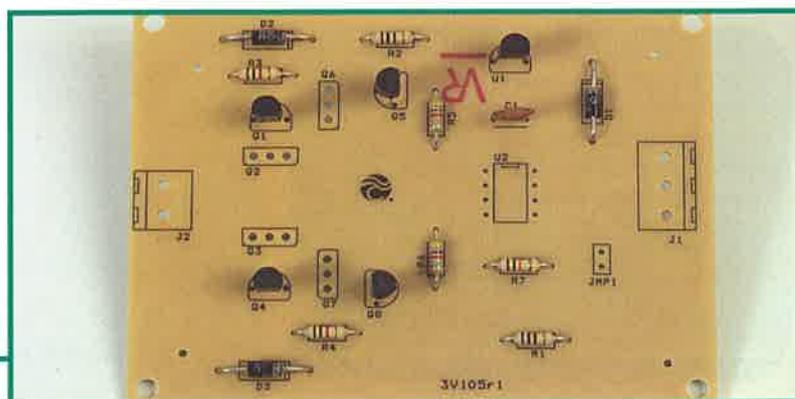
Il diodo successivo è l'ultimo da saldare sulla scheda di controllo della pinza: è un altro 1N4007, che andrà in D3. Come sempre, dobbiamo prestare attenzione all'inserzione, per la polarità. Taglieremo la restante parte del reoforo di ogni diodo.



Salderemo ora un regolatore di tensione 78L05 in U1. Servirà per abbassare la tensione di alimentazione della scheda da 9 a 5 V. Questo regolatore ha un contenitore tipo TO92, al posto del TO220 del 7805 della scheda di controllo. Dobbiamo inserirlo come qualsiasi altro transistor, secondo la mezza luna disegnata nella serigrafia della scheda.



Questa è l'immagine finale della scheda di controllo della pinza, con quasi tutti i componenti già saldati. Dobbiamo verificare di non aver equivocato né il posto di inserzione dei componenti, né il loro orientamento.



```
c:\progra~1\mplab\asc_55\eserciz.asm
1
2      List    p=16F84A
3      include "P16F84a.INC"
4
5      org    0x00
6      goto  INIZIO
7      org    0x05
8
9 INIZIO:      bsf    STATUS, RP0
10           clr   PORTA
11           movlw b'01110000'
12           movwf PORTB
13           bcf   STATUS, RP0
14           clr   PORTA
15           bcf   PORTB, 7
16
```

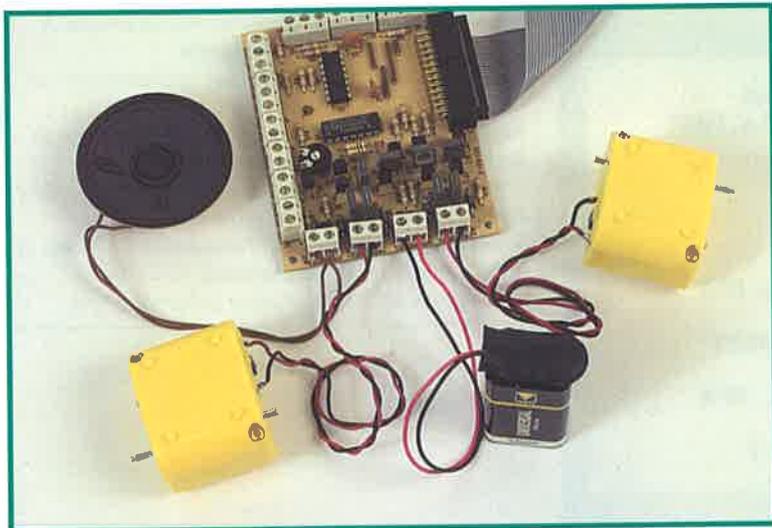
Nel seguente esercizio combineremo la scheda di potenza, quella dei sensori e quella delle luci. L'esercizio consiste nell'implementare un robot guardiano. Quando si fa notte, il robot inizia a girare all'interno dell'abitazione, in perlustrazione alla ricerca di rumori sospetti, fermandosi quando sente un rumore e rimanendo in attesa per rilevare qualche movimento. Se rileva movimento, attiva l'altoparlante a modo di allarme. Nell'immagine si osserva la sequenza abituale dell'inizio del programma.

Utilizzeremo tre ingressi digitali della porta B: RB5 per collegare gli ultrasuoni (rivelazione di movimento), RB4 (rivelatore giorno/notte) per il sensore di luce, e RB6 per il rivelatore di suono. Come uscita per la gestione dei motori configureremo la porta A. All'inizio spegneremo l'altoparlante che si trova su RB7, e fermeremo i motori: il robot è fermo. Il robot inizia a muoversi quando il segnale del sensore di luce indica che è notte. Quando il robot è in movimento e si produce un suono, un ordine di fermata spegnerà i motori e il robot si ferma.

```
:\progra~1\mplab\asc_55\eserciz.asm
16
17 ATTENDI_NOTTE:  btfss  PORTB, 4
18                goto   ATTENDI_NOTTE
19                movlw  b'00000101'
20                movwf  PORTA
21
22 CERCA_SUONO:    btfss  PORTB, 6
23                goto   CERCA_SUONO
24                clr    PORTA
25
```

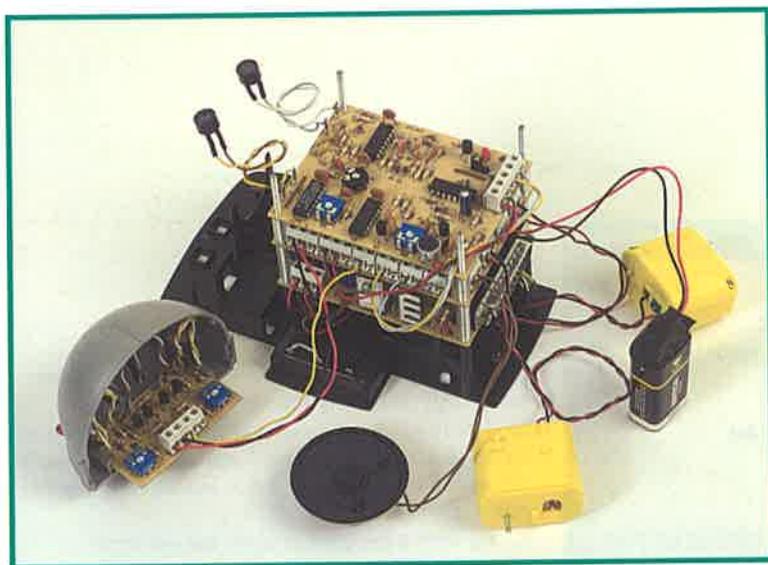
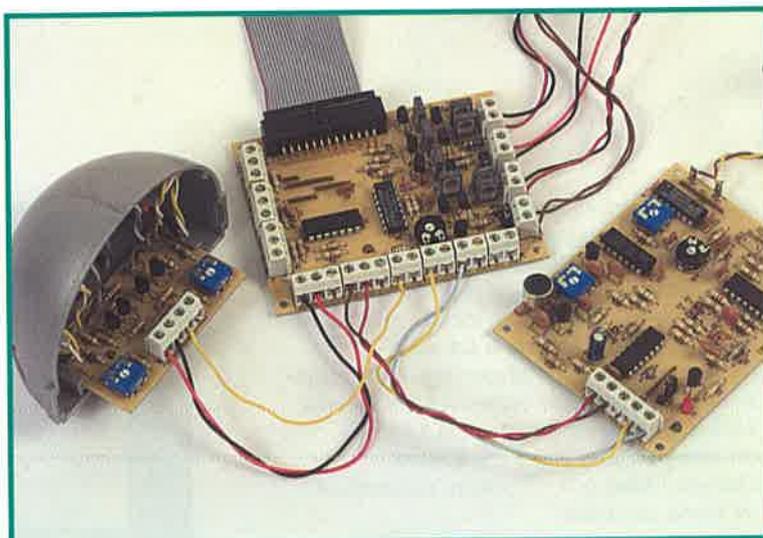
```
c:\progra~1\mplab\asc_55\eserciz.asm
25
26 ATTENDI_MOVIMENTO:
27                btfsc  PORTB, 5
28                goto   ATTENDI_MOVIMENTO
29
30 ALLARME_ON      bsf    PORTB, 7
31                goto   ALLARME_ON
32
33                END
```

Da questo momento attenderemo un segnale che arriva dagli ultrasuoni tramite l'ingresso RB5. Se si rileva un movimento si attiverà l'altoparlante in modo allarme, inviando un 1 sull'uscita RB7. Questa situazione si mantiene in modo indefinito. L'utente può e deve modificare il programma per adattarlo alle sue necessità.



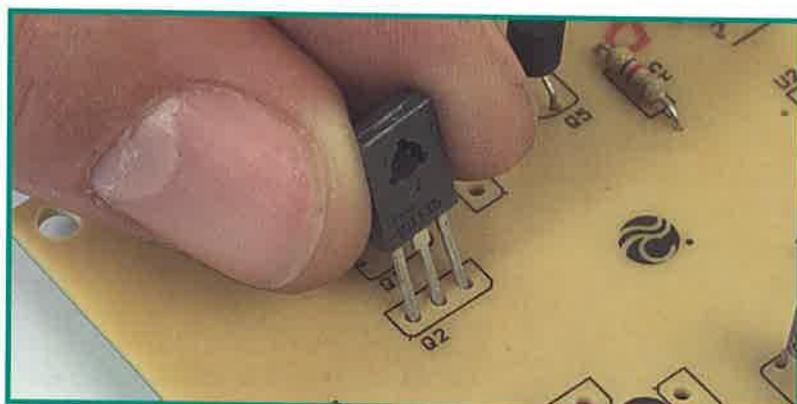
Questi sono i collegamenti che dobbiamo fare con la scheda di potenza. I due motori si collegano in J11 e J12, e la tensione di alimentazione degli stessi nel morsetto J13. L'altoparlante si collegherà tramite il morsetto J10. Dobbiamo tenere il jumper JMP1 chiuso, per fare in modo che l'altoparlante possa suonare.

Dobbiamo collegare le due schede dei sensori tramite la scheda di potenza. Utilizzeremo i morsetti J6, J7 e J8, che sono quelli ancora liberi sulla scheda di potenza, per il collegamento dei sensori esterni. I morsetti J6, J7 e J8 fanno riferimento rispettivamente ai pin RB4, RB5 e RB6 del microcontroller PIC. In J6 introdurremo il segnale di luce che arriva dal pin 3 (OUT1) del morsetto JMP1 della scheda dei sensori di luce. In J7 il segnale di ultrasuoni che si ottiene dal pin 4 del morsetto J1 della scheda dei sensori. Infine in J8 collegheremo i segnali del rivelatore di suono che è disponibile sul pin 5 di J1 della scheda dei sensori.



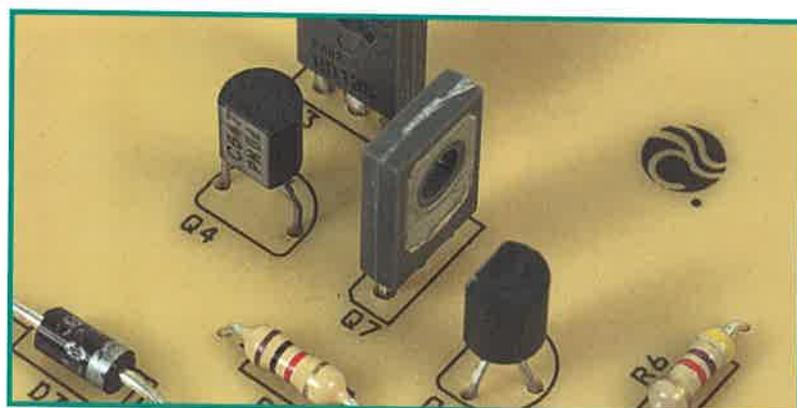
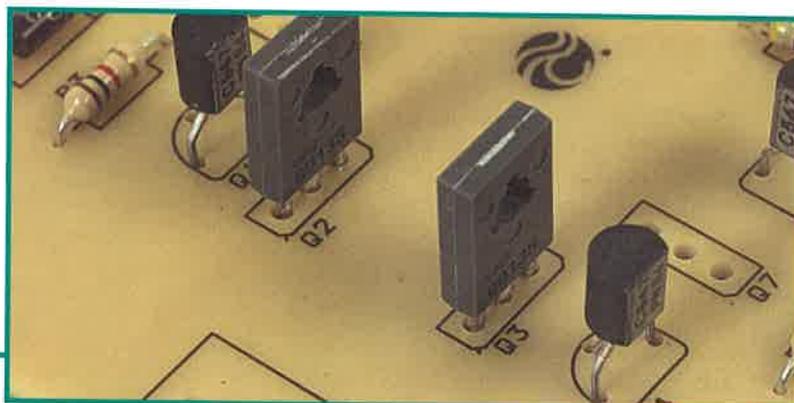
In questa immagine si mostrano tutte le schede collegate e il circuito in funzione. Tramite i sensori dobbiamo provocare una sequenza di segnali di ingresso appropriata, per verificare che l'esercizio funzioni correttamente. Regoleremo la sensibilità del microfono con R30 per fare in modo che sia bassa, e non sia influenzata dal rumore dei motori in funzione. Si propone al lettore di ripetere l'esercizio, facendo però in modo che il robot segua, mediante due sensori ottici, una linea nera sul pavimento (che chiuderà un circuito), mentre ricerca suoni sospetti.

Montaggio passo a passo



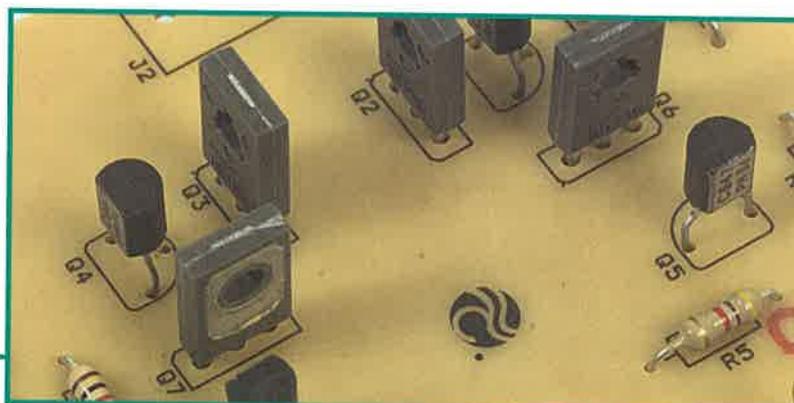
Collocheremo un transistor BD135 in Q2. È importante inserire il transistor nel verso e nell'orientazione mostrati nell'immagine.

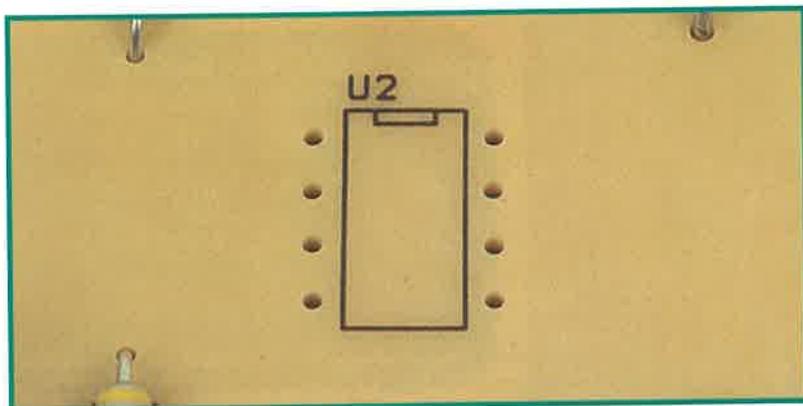
Dobbiamo collocare un altro transistor BD135 in Q3. I transistor non devono essere introdotti a filo della scheda, ma devono rimanere sollevati di qualche millimetro sulla sua superficie.



Un altro transistor BD135 verrà inserito in Q7 secondo il verso indicato nell'immagine, che si può riconoscere dalla forma della serigrafia disegnata sulla scheda.

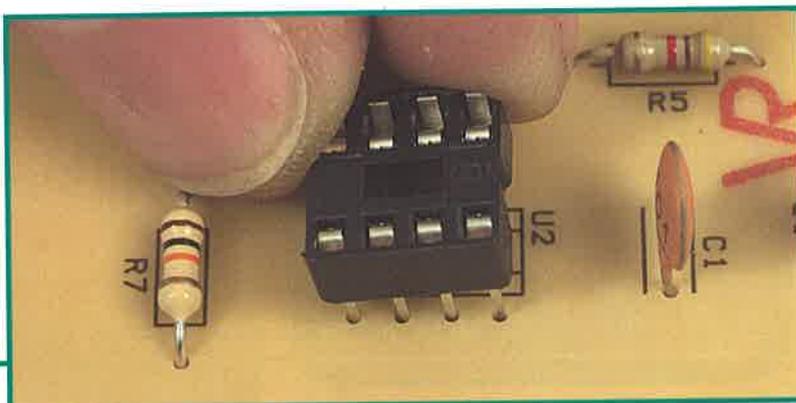
In Q6 collocheremo l'ultimo transistor di questa scheda. È un altro BD135 e va nel verso mostrato. Dopo aver saldato questi transistor abbiamo terminato il ponte ad H per il controllo del motore.



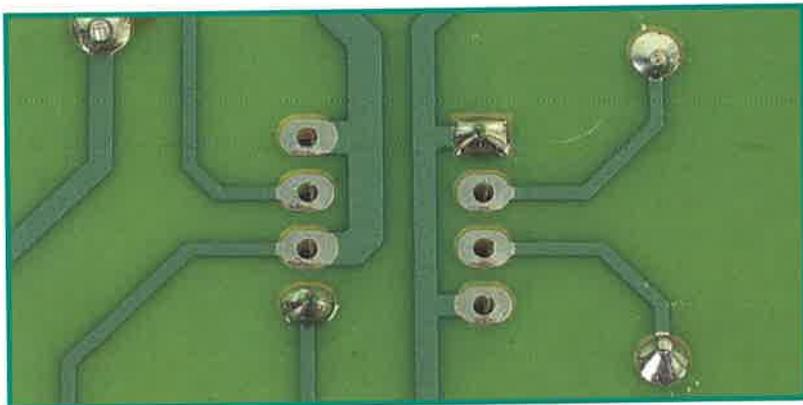


Dobbiamo montare uno zoccolo da 8 pin in U2, per poi posizionargli sopra un microcontroller PIC, programmato per il controllo del movimento di apertura e chiusura della pinza.

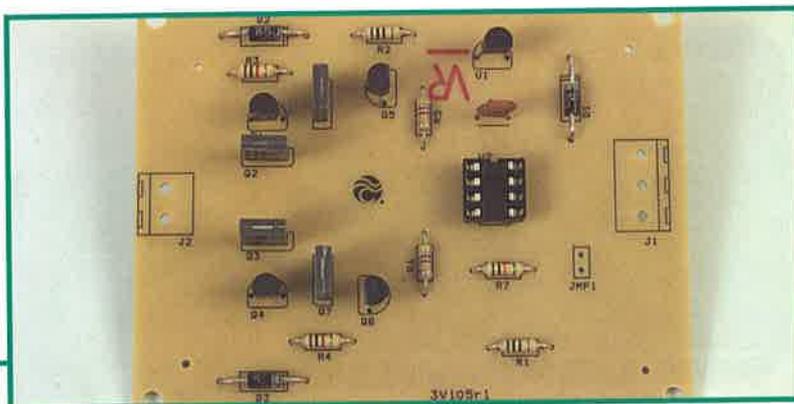
Lo zoccolo deve essere inserito tenendo conto del verso indicato dalla tacca di riferimento disegnata nella serigrafia della scheda, e presente anche sullo zoccolo, che indicherà il verso di inserzione del circuito integrato.



Inizialmente salderemo solo due piedini agli estremi dello zoccolo. Dopo aver realizzato queste due prime saldature, dobbiamo verificare che lo zoccolo sia a filo della scheda e correttamente inserito.



Questo è l'aspetto della scheda di controllo della pinza con i componenti inseriti fino a questo momento. È già praticamente terminata e presto potremo controllare con essa la pinza del robot.



Il robot giardiniere

```

c:\progra~1\mplab\asc_56\eserc.asm
1
2      List      p=16F84A
3      include  "P16F84A.INC"
4
5  CONTATORE: EQU      0x0c
6
7      org      0x00
8      goto    INIZIO
9      org      0x05
10
11 INIZIO:      bsf     STATUS, RP0
12             clr    PORTA
13             movlw  b'00111100'
14             movwf  PORTB
15             movlw  0x07
16             movwf  OPTION_REG
17             bcf    STATUS,RP0
18             clr    PORTA
19

```

In questo esercizio combineremo la scheda di controllo, quella di potenza, e quella dei sensori di luce, con l'obiettivo di implementare un robot tosaerba. Il robot attenderà che sia giorno per iniziare il suo lavoro. Possiede due sensori meccanici mediante i quali rileva le collisioni. Quando urta, gira e continua il suo cammino. Con un battere di mani, o un suono forte, possiamo parlare al robot. Nell'immagine è mostrata la sequenza tipica di inizio.

Configureremo la porta A come uscita per la gestione dei motori. I due sensori meccanici incaricati di rilevare le collisioni del robot si trovano in RB2 e RB3. Il rilevatore di livello luminoso che indica al robot quando è giorno si trova in RB4, e il rilevatore sonoro per fermare il robot in RB6. Quando sarà giorno inizieremo a muovere il robot controllando che non urti e non si producano dei suoni che lo fermino.

```

c:\progra~1\mplab\asc_56\eserc.asm
20 ASPETTA_GIORNO: btfss  PORTB, 4
21                goto   ASPETTA_GIORNO
22
23 TAGLIA_PRATO:  btfsc  PORTB, 2
24                goto   URTO
25                btfsc  PORTB, 3
26                goto   URTO
27                movlw  b'00001001'
28                movwf  PORTA
29                btfsc  PORTB, 5
30                goto   SUONO
31                goto   TAGLIA_PRATO
32

```

```

c:\progra~1\mplab\asc_56\eserc.asm
32
33 SUONO:         clr    PORTA
34                goto   SUONO
35
36 URTO:          movlw  b'00001001'
37                movwf  PORTA
38                call   DELAY
39                goto   TAGLIA_PRATO
40

```

Se si produce un suono come un battere di mani, il robot si arresterà e rimarrà fermo fino a che resetteremo il sistema. Possiamo regolare la sensibilità del microfono con il corrispondente potenziometro della scheda dei sensori. Se il robot urta qualcosa, faremo in modo che giri su di un lato per un secondo. Trascorso questo tempo torneremo al ciclo principale del programma. Se l'ostacolo è stato schivato il robot continuerà il suo cammino, altrimenti continuerà a girare.

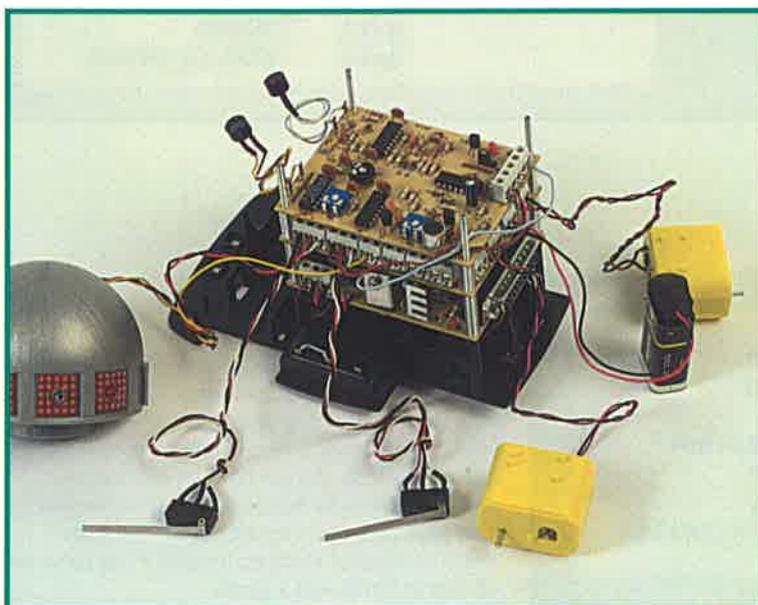
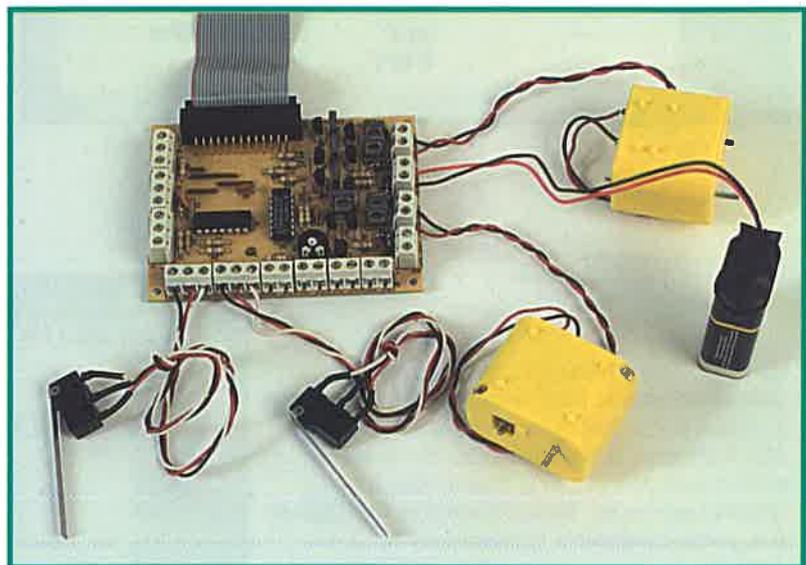
```

c:\progra~1\mplab\asc_56\eserc.asm
40
41 DELAY:      movlw   .100
42            movwf   CONTATORE
43 DELAY10MS:  movlw   0xD8
44            movwf   TMR0
45 DEL        btfss   INTCON, 2
46            goto   DEL
47            bcf     INTCON, 2
48            decfsz  CONTATORE, 1
49            goto   DELAY10MS
50            return
51
52            END

```

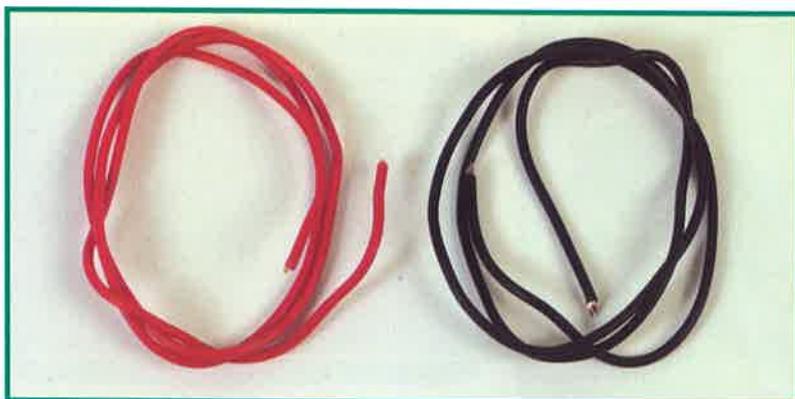
Questa è una routine di Ritardo di 1 secondo, che consiste nel chiamare 100 volte una routine di temporizzazione di 10 millisecondi. Lavora con un prescaler da 256 che è stato configurato all'inizio del programma nel registro OPTION. Durante il tempo di attesa il microrobot continua a girare al fine di evitare l'ostacolo.

Questi sono i collegamenti dei motori e dei sensori meccanici (fine corsa) alla scheda di potenza, nelle morsettiere J3 e J4 (RB2 e RB3). Inoltre dobbiamo introdurre tramite il morsetto J13 la tensione di alimentazione per i motori, che collegheremo in J11 e J12. L'immagine ci mostra questi collegamenti.



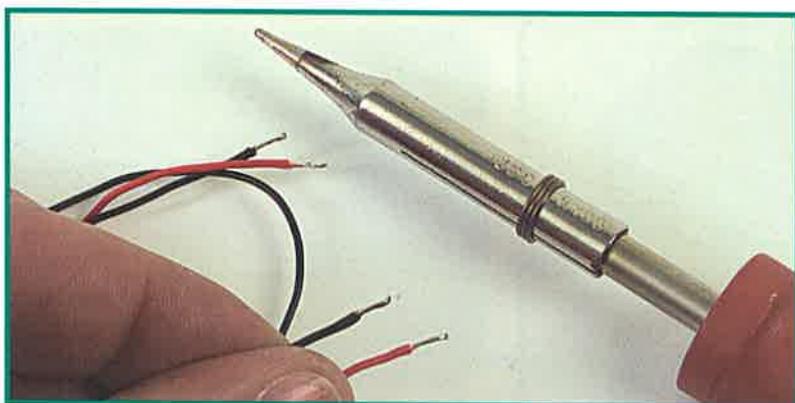
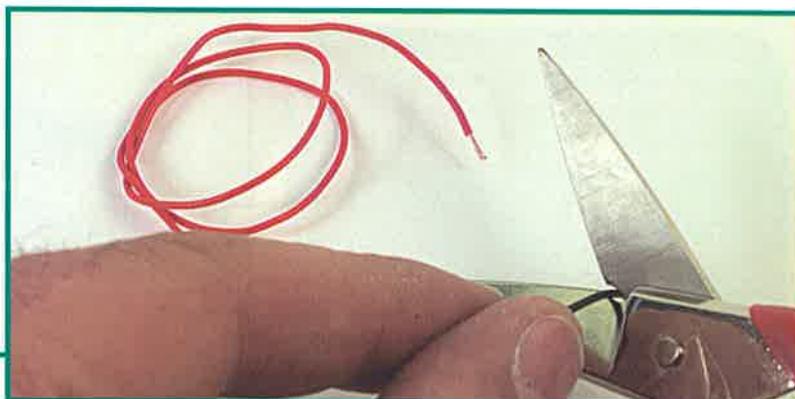
Questa è un'immagine globale di tutto l'esercizio, collegato e funzionante. Dobbiamo collegare la scheda delle luci e dei sensori mediante la scheda di potenza. Da un lato metteremo l'alimentazione per entrambe le schede, oltre agli opportuni segnali di ingresso. Il segnale del sensore delle luci, pin 3 di JMP1 della scheda dei sensori, entrerà tramite il pin 2 del morsetto J6 (RB4), e il segnale del sensore di suono, pin 4 di J1 della scheda dei sensori, entrerà sul pin 2 del morsetto J7 (RB5) della scheda di potenza.

Montaggio passo a passo



Prepareremo questi due pezzi di filo per alimentare la scheda di controllo della pinza tramite i corrispondenti morsetti della scheda di potenza. Il rosso servirà per portare alimentazione da 9 V, quello nero per la massa.

Peleremo gli estremi di ogni cavo di circa 5 mm. Per fare ciò utilizzeremo delle forbici o uno spelafili. Bisogna fare attenzione a tagliare solo la guaina di protezione e non stringere troppo, per non danneggiare il conduttore di rame. Qualunque torsione o tensione successiva su questo cavo, può causarne la rottura.



Questi cavi, una volta pelati, devono essere intrecciati e prestagnati, come si vede nella figura. In questo modo eviteremo che si sfilino e ne faciliteremo il loro successivo collegamento ai morsetti corrispondenti.

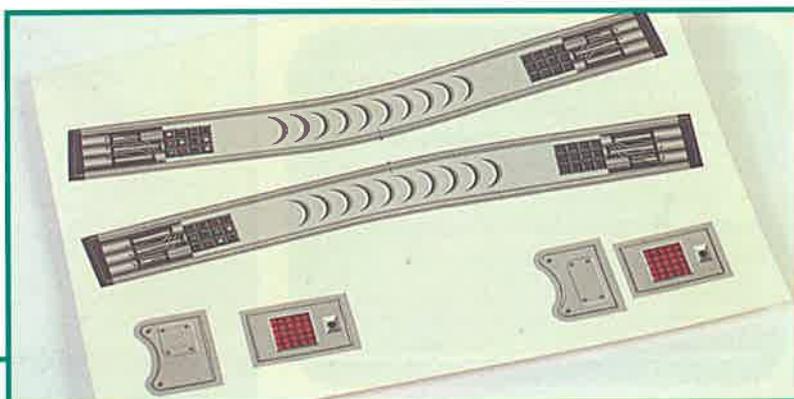
Le sei viti autofilettanti mostrate nell'immagine si utilizzeranno al momento opportuno per chiudere i pezzi del braccio di Monty.





Queste sono le due braccia che formano parte della struttura, o corpo, di Monty. Ora ne eseguiamo la decorazione.

L'immagine ci mostra un foglio di dimensione A4 con gli adesivi per decorare le braccia di Monty. Questi adesivi sono stati pre-incisi per poter togliere le lamine che ci interessano con facilità.

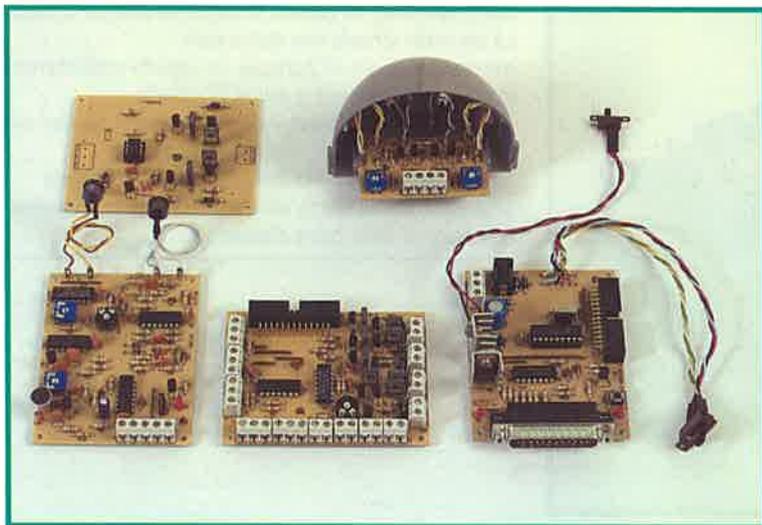


Prima di collocare gli adesivi dobbiamo essere sicuri della pulizia della superficie dove li incolleremo. Li staccheremo con attenzione dal foglio e li posizioneremo come indicato nella fotografia.

Nell'immagine è riportato l'aspetto finale di entrambe le braccia debitamente decorate. Si deve fare molta attenzione al momento di incollare gli adesivi lunghi che ornano il contorno del braccio. Eseguite l'operazione lentamente, in modo che non rimangano bolle d'aria.

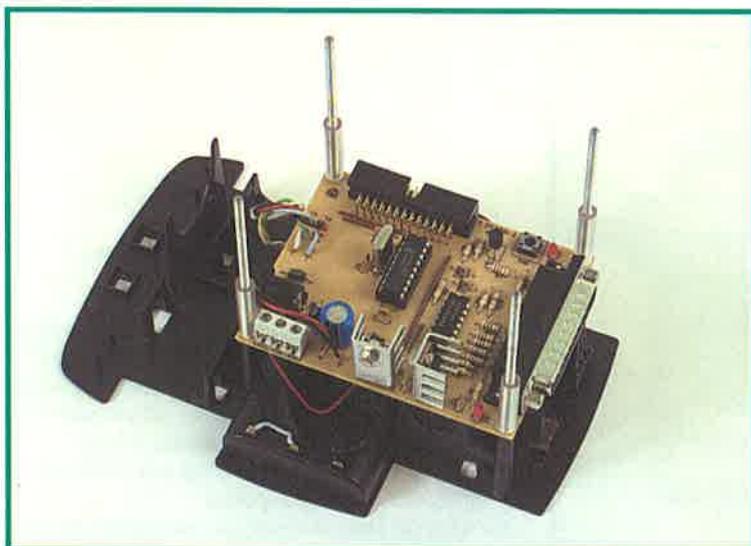
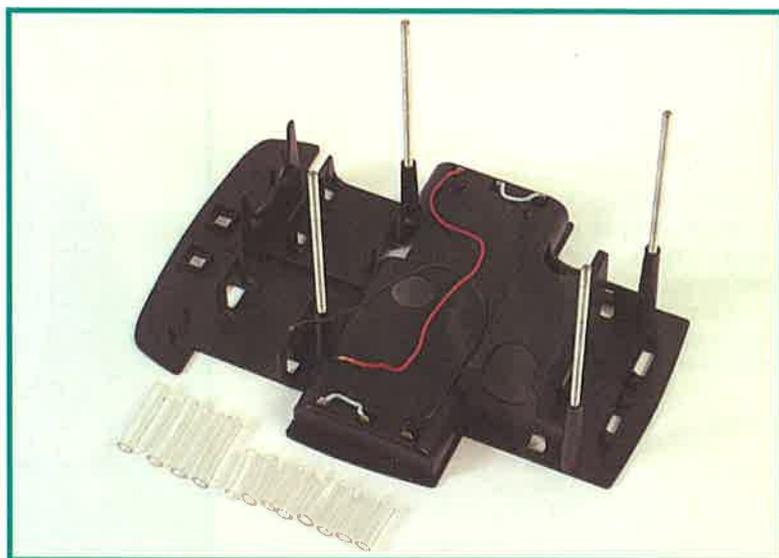


Montaggio finale

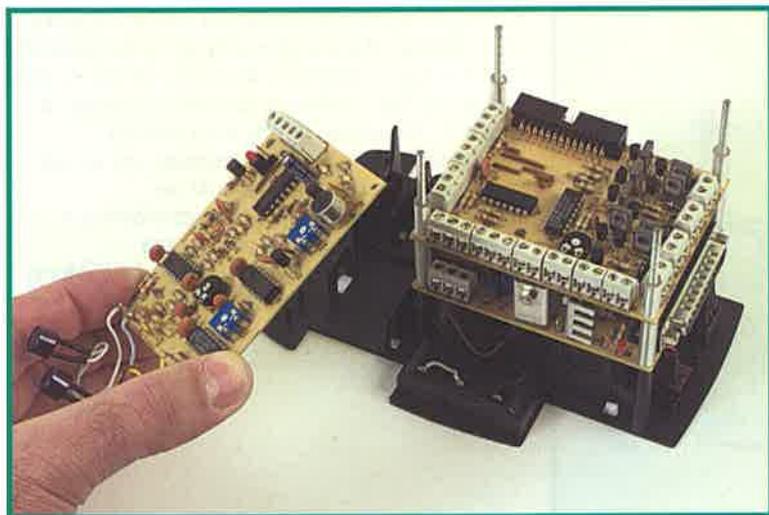


In questo capitolo offriremo una visione d'insieme delle diverse schede elettroniche di cui è composto il robot, e la loro corretta collocazione all'interno dello stesso. Il robot contiene le schede di controllo, di potenza, dei sensori, di luce e di controllo della pinza. Inoltre abbiamo montato una scheda di ingressi e uscite che ci è servita per imparare a programmare il microcontroller, e a realizzare gli esercizi base, però questa scheda non fa parte della struttura finale di Monty.

Sul telaio del motore dobbiamo inserire le quattro barre filettate da 3 mm come si vede nell'immagine; queste barre vi sono state fornite tempo fa. Su queste barre monteremo, come una torre, le diverse schede del robot mediante le quali si realizzano tutte le funzioni. Vi è stato fornito, inoltre, un tubo di PVC che, debitamente tagliato in pezzi, permette di separare le diverse schede fra loro. Le dimensioni dei pezzi intermedi di tubo saranno: 4 da 25 mm (indicati in PL 48) e 8 da 16 mm.

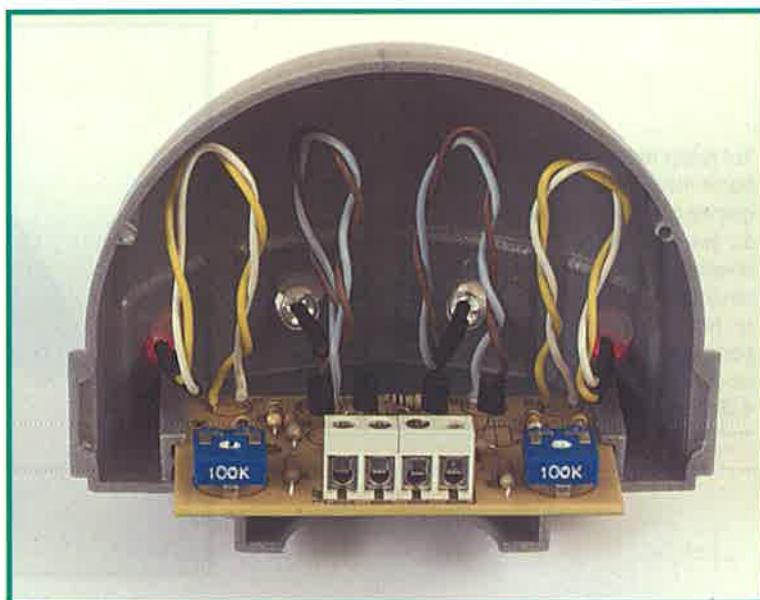


La prima scheda che monteremo sul telaio del robot è quella di controllo. Deve essere incastrata sulla base del medesimo, inserendo successivamente i 4 tubi da 25 mm, così come è mostrato nell'immagine. È importante che questa prima scheda sia ben posizionata, dato che dispone di diversi connettori verso l'esterno come il DB25 per la porta parallela o il connettore di alimentazione.



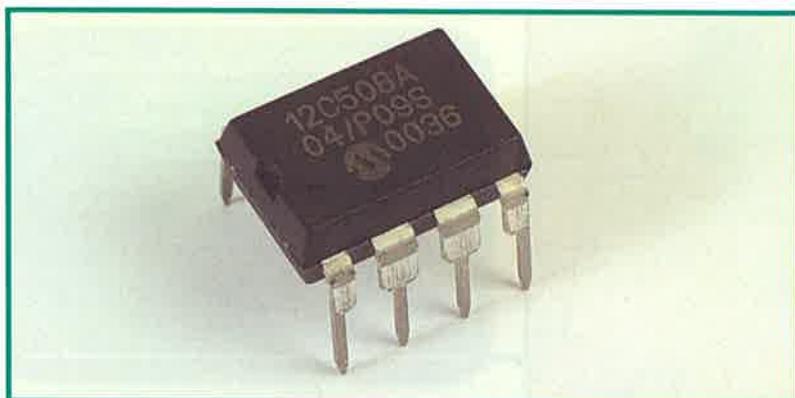
Ora inseriremo su questa struttura le diverse schede. La seconda scheda che dobbiamo montare è quella di potenza. In seguito collocheremo la scheda dei sensori e infine la scheda di controllo della pinza. I pezzi di plastica tagliati alla lunghezza di 16 mm saranno introdotti nelle quattro barre filettate, fra scheda e scheda. Una volta terminata la struttura avviteremo i bulloni agli estremi delle barre filettate.

La scheda delle luci non va montata nella struttura principale del robot, ma all'interno della testa di Monty. Deve essere incastrata come si vede nell'immagine, ben posizionata, e con i due diodi e i due sensori ottici inseriti nei fori della testa del robot, come vi abbiamo spiegato in precedenza.



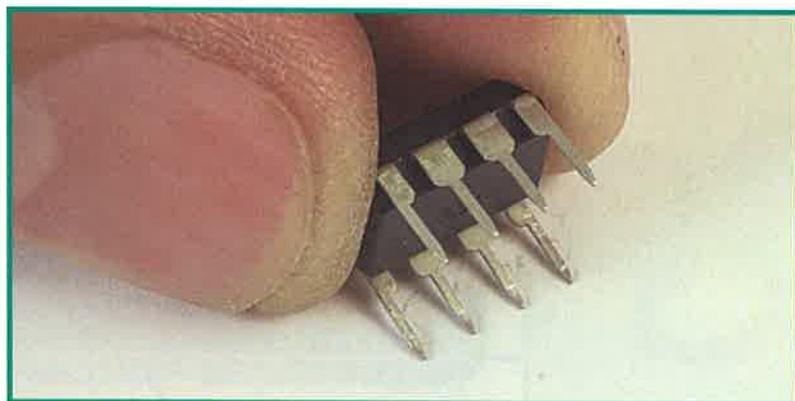
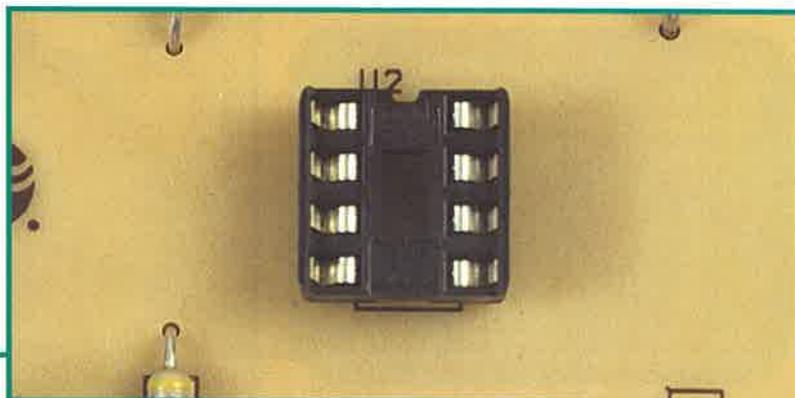
Nell'immagine si mostrano le due metà che formano il corpo di Monty. Come già spiegato nei fascicoli precedenti, nei punti previsti e su entrambe le metà, bisognerà aver realizzato le corrispondenti aperture che danno accesso ai connettori della scheda di controllo, e permettono di alloggiare le capsule ad ultrasuoni. Inoltre si suppone che siano già state decorate con gli appositi adesivi.

Montaggio passo a passo



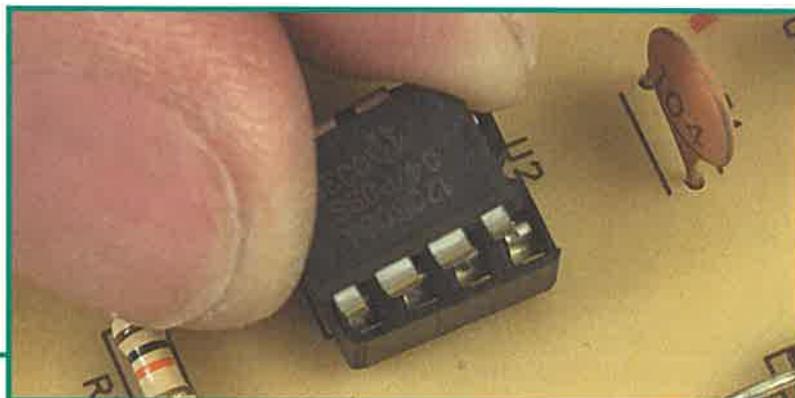
Questo chip è un microcontroller PIC modello 12C508A, che ha il compito di controllare il motore della pinza del braccio del robot. Questo PIC è stato programmato per realizzare le manovre di apertura e chiusura della pinza, quando riceve gli opportuni segnali tramite il PIC16F84 principale.

È molto importante inserire il microcontroller sul suo zoccolo nella posizione corretta, altrimenti al momento di accendere la scheda potremmo danneggiare il chip. Il verso di inserzione è indicato dalla tacca posizionata sulla parte superiore dello zoccolo, com'è mostrato nell'immagine.



È necessario allineare leggermente i piedini del chip prima di inserirlo nello zoccolo. Per fare questo faremo pressione su una superficie liscia, come un tavolo, come si vede nell'immagine. Quando i piedini formeranno un angolo di 90° rispetto al corpo del chip saranno pronti, e procederemo all'inserzione.

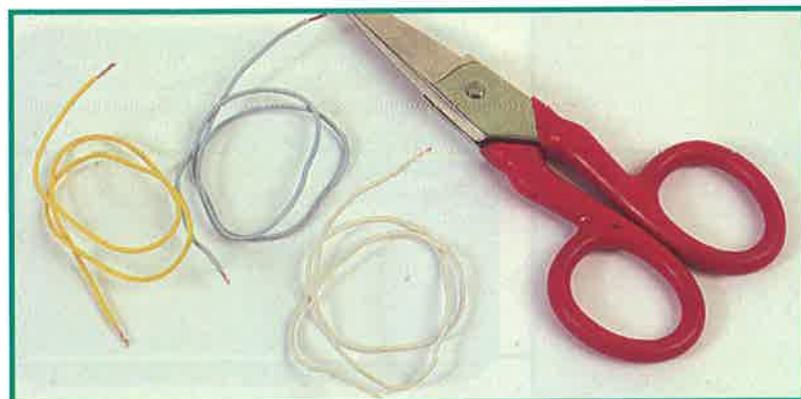
Il verso del chip è indicato da una tacca di riferimento, così come sullo zoccolo che abbiamo saldato sulla scheda. Dovremo fare coincidere la tacca dello zoccolo con quella del microcontroller.





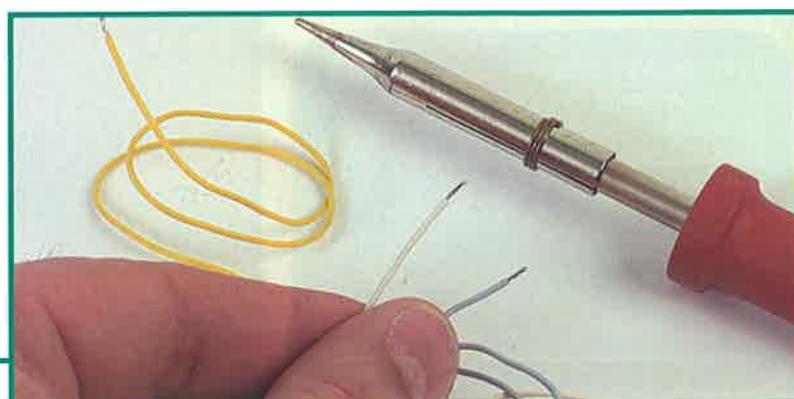
Inseriremo il circuito integrato nello zoccolo sino a che sarà totalmente introdotto, come si vede nell'immagine. Altrimenti, il chip non farebbe un buon contatto e con il passare del tempo potrebbe provocare un malfunzionamento.

Questi tre cavi serviranno per alimentare la scheda di controllo della pinza, e per inviare gli ordini che controllano l'apertura e la chiusura della medesima.



Prepareremo ora questi cavi. Li speleremo ad entrambe le estremità, utilizzando uno spelafili o un paio di forbici da elettricista. Dobbiamo fare attenzione a non tagliare troppo, per non danneggiare il conduttore interno.

Conviene tagliare e prestagnare le punte dei cavi, per fare in modo che quando li introdurremo nei morsetti rimangano ben serrati e non si staccino, evitando così la possibilità di cortocircuiti.

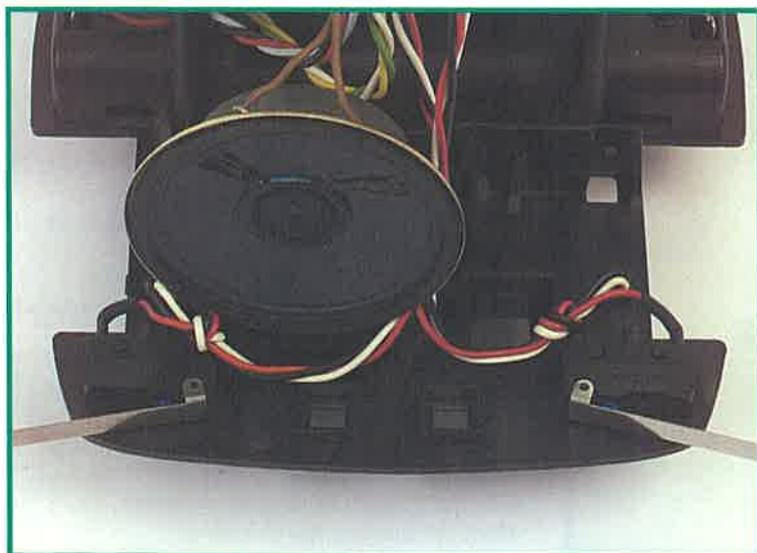


Montaggio finale

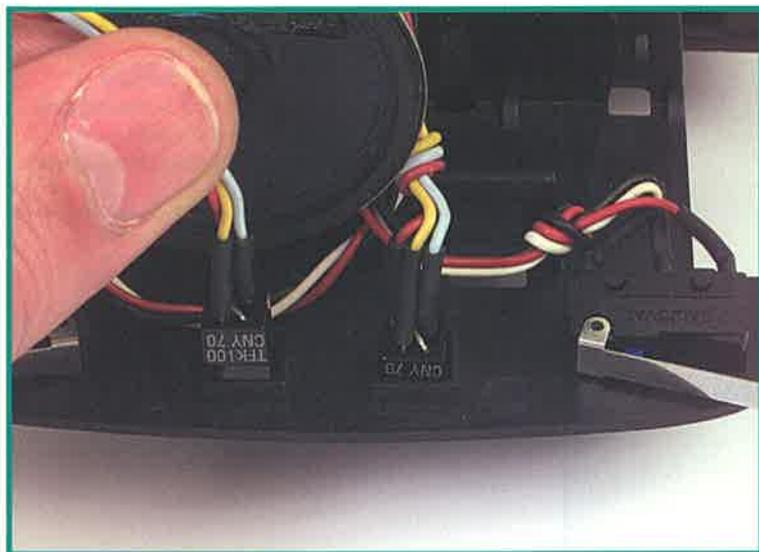


Come si vede nell'immagine, la parte posteriore del corpo di Monty permette l'accesso al connettore DB25 della scheda di controllo. Tramite questo connettore si realizza la connessione con il PC, in modo da poter programmare sul PIC i numerosi programmi di applicazione senza aprire la struttura del robot.

Per fare in modo che entrambi i pezzi del corpo di Monty restino uniti in modo sicuro fra loro, è necessario incastrarli debitamente sul telaio e utilizzare le clips che sono state fornite nei numeri precedenti. Nell'immagine si mostra una clip che unisce le due metà del corpo.

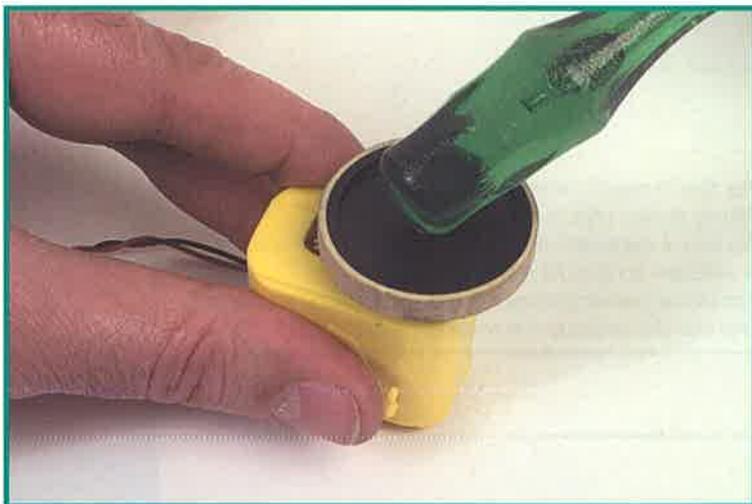


I sensori meccanici o fine corsa vanno montati sul telaio del robot, con i terminali posizionati come mostrato nell'immagine. Questi sensori si incastrano sui due terminali di plastica di cui dispone il telaio, dove resteranno perfettamente fissati se verrà fusa la parte superiore. Nell'immagine, inoltre, si può vedere il posizionamento dell'altoparlante che sarà fissato con una goccia di adesivo. Vi è stato fornito un pezzo di cavo da 40 cm, che tagliato in due, verrà saldato sull'altoparlante nel caso in cui quelli attuali siano troppo corti.

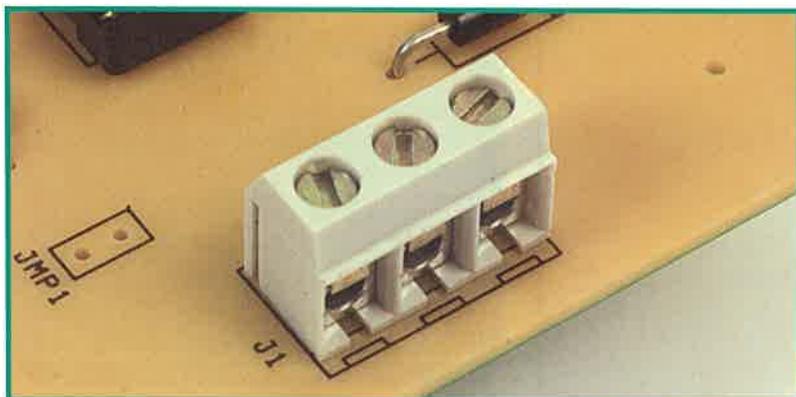


I due sensori ottici a riflessione che servono per controllare la direzione del robot, vanno anch'essi fissati sul telaio. Sulla parte anteriore del telaio, a lato dei fine corsa, si trovano due aperture quadrate su cui verranno incastrati questi sensori. Li introdurremo e li fisseremo. Potremo mettere qualche goccia di adesivo per maggiore sicurezza. Si può migliorare la risposta del robot limando l'apertura per avvicinare i sensori al suolo.

Queste sono le ruote di trazione del robot. Disponiamo di due gomme che sono montate sulle ruote come un pneumatico. Nell'immagine si mostra il montaggio delle ruote su uno degli assi dei motori. Vanno inserite a pressione. Potremo battere leggermente (ad esempio col manico di un cacciavite) sul centro della ruota perché si incastrino perfettamente, appoggiando l'altro estremo dell'asse su una superficie dura.

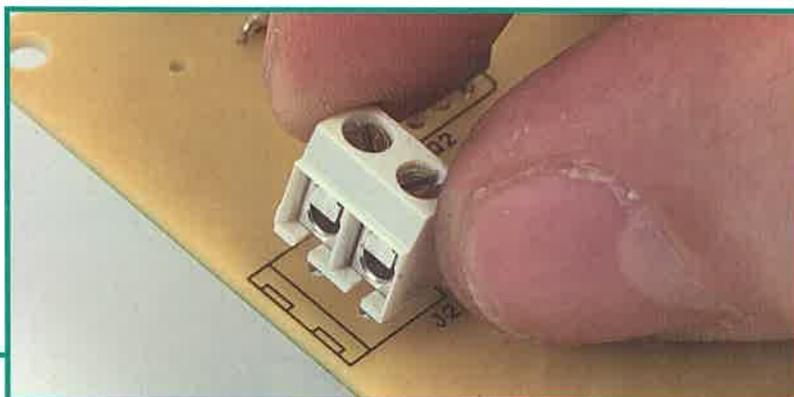


Nell'immagine possiamo vedere i fori praticati sui lati inferiori della metà anteriore del corpo di Monty. Si realizzano sulla zona predisposta, utilizzando un oggetto appuntito come abbiamo già fatto altre volte. Con una lima si eliminano le sbavature che potrebbero rimanere. Tramite questa apertura fuoriescono i fine corsa o sensori meccanici. Inoltre si può vedere come le due capsule a ultrasuoni appaiono sulla stessa metà anteriore del corpo di Monty. Queste capsule possono essere fissate con qualsiasi tipo di adesivo.



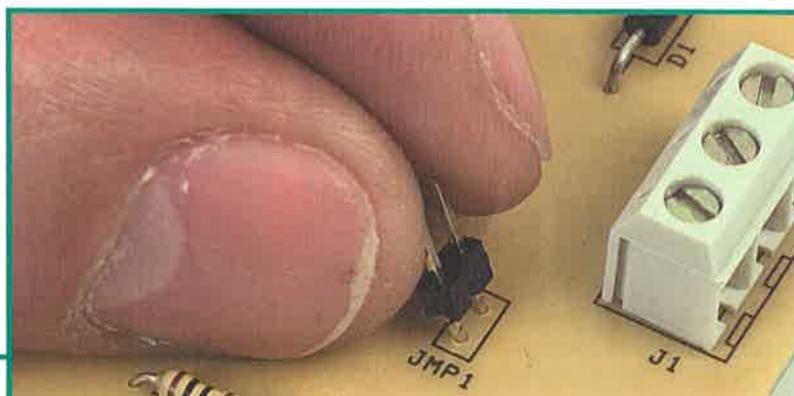
Salderemo ora gli ultimi componenti della scheda per il controllo della pinza. Monteremo una morsettieria da tre contatti su J1. Questi contatti devono rimanere verso l'esterno della scheda, come mostrato nell'immagine.

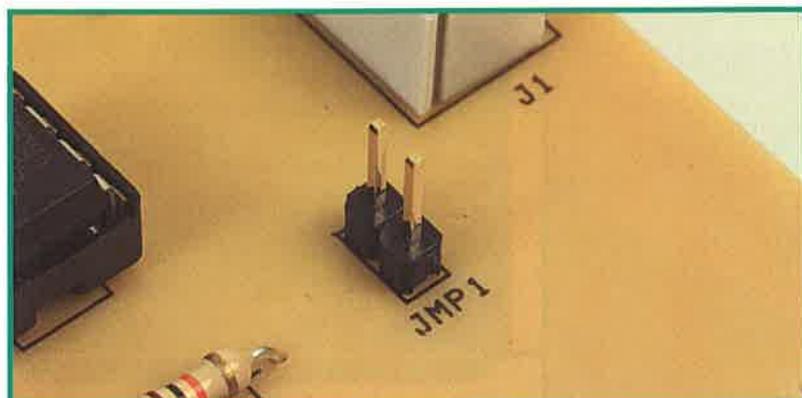
Ora salderemo un'altra morsettieria in J2. È una morsettieria a due contatti, che devono rimanere orientati verso l'esterno della scheda. La morsettieria deve essere introdotta fino a rimanere a filo della scheda.



Nell'immagine si possono vedere le saldature realizzate sulla morsettieria. È necessario utilizzare più stagno rispetto alle saldature del resto dei componenti elettronici della scheda. Dobbiamo ottenere dei coni brillanti e appoggiati sulla superficie.

Dobbiamo ora saldare il connettore a due vie dritto in JMP1. Inseriremo nella scheda la parte corta dei piedini. La parte lunga che sporge dalla scheda servirà per introdurre un jumper mediante il quale apriremo o chiuderemo una linea elettrica.



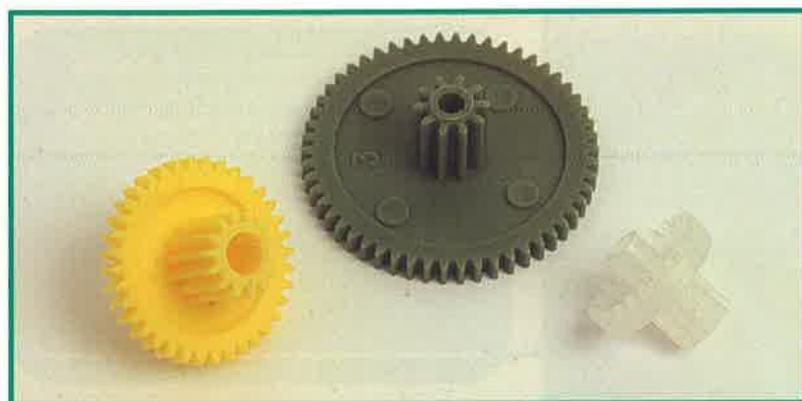


Dobbiamo saldare il piccolo connettore a due vie in modo che resti perfettamente dritto, come si vede nell'immagine. Facciamo in modo che questo non rimanga storto, inclinato o diseguale in altezza.

Una volta saldati i due piedini ci assicuriamo della corretta inserzione e monteremo il jumper a due contatti che serve per chiudere la linea ogni volta che è inserito.



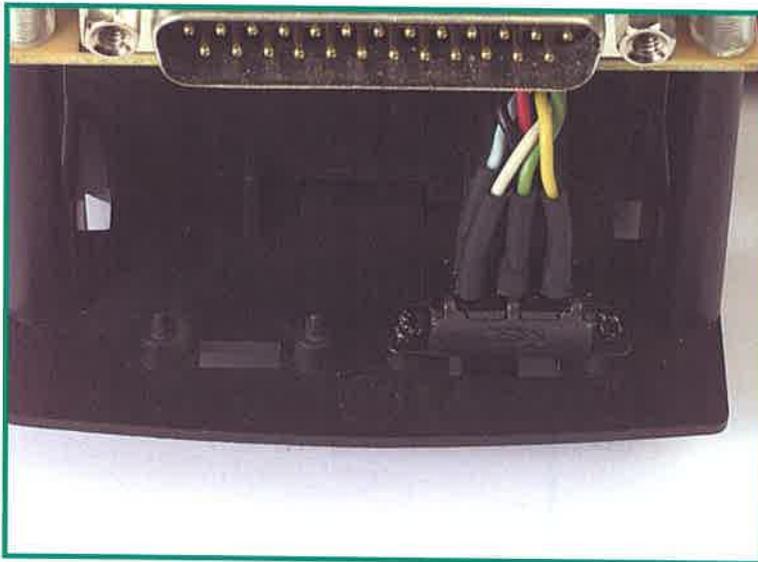
Nell'immagine si possono vedere due ingranaggi e una corona. Questi elementi meccanici vanno all'interno del braccio di Monty e hanno il compito di produrre l'apertura e la chiusura della pinza mediante un movimento di rotazione dell'asse del motore.



Questa è una delle braccia di Monty. Su di esso monteremo gli ingranaggi e il motore per l'apertura e chiusura della pinza. Per prima cosa, con l'aiuto di un cutter, dobbiamo tagliare la parte predisposta all'estremo del braccio. Su questo estremo fisseremo in seguito le pinze.

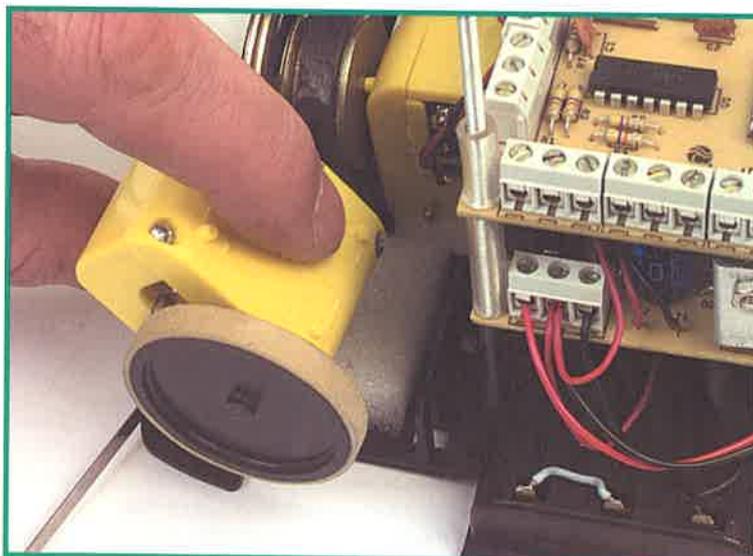
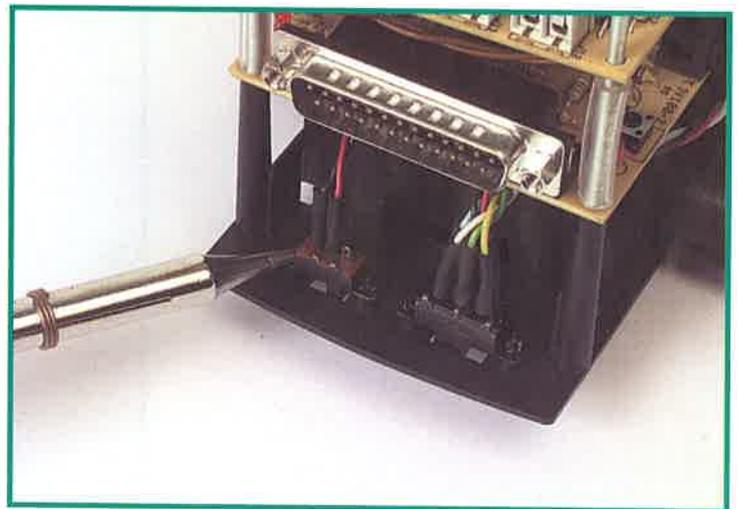


Montaggio finale

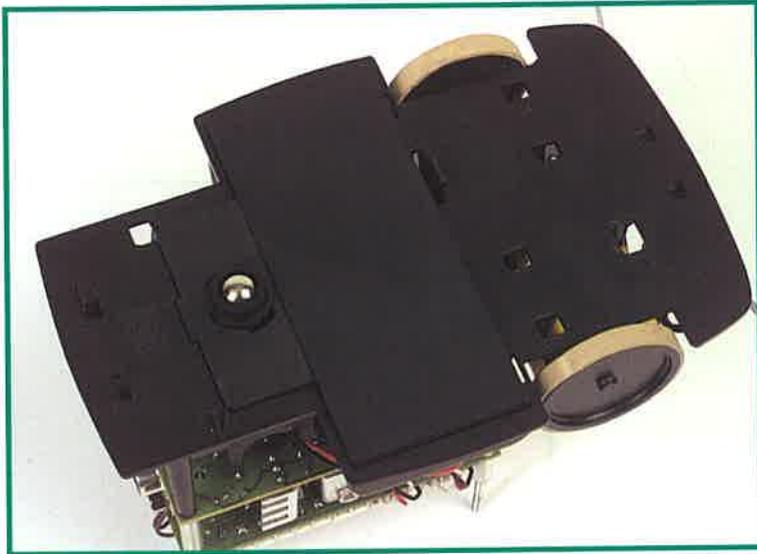


Ci sono due interruttori che dobbiamo fissare al telaio di Monty. Uno è quello che ci permette di selezionare tra modo scrittura e modo Run; è il più grande e va incastrato in uno dei due fori nella parte posteriore del telaio, così come si vede nell'immagine.

L'altro interruttore serve per collegare o scollegare l'alimentazione del robot ed è situato vicino all'interruttore di scrittura. Una volta montati entrambi gli interruttori, possiamo riscaldare con il saldatore il supporto di plastica su cui sono sistemati per fare in modo che restino fissati al telaio in modo definitivo.

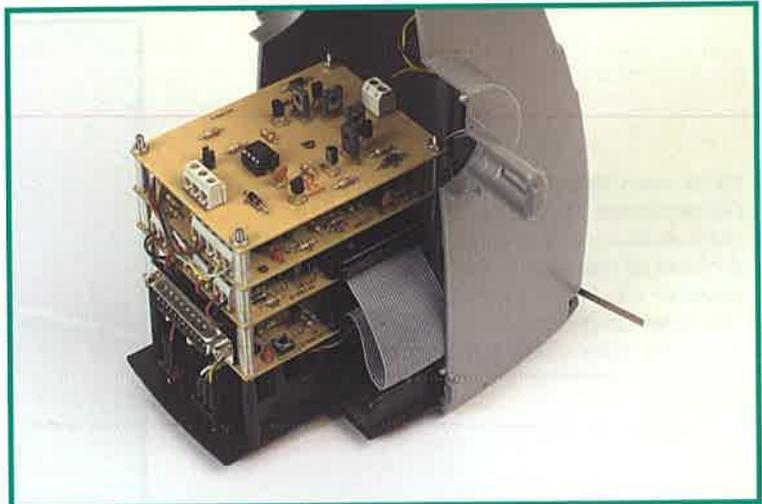


Qui si incastrano i due motori che spingono Monty, con i loro rispettivi telai. Fra i contenitori dei motori e il telaio di Monty sono posti, con funzione di ammortizzatori, due pezzi di gomma piuma. È molto importante che i motori siano ben incastrati nella loro posizione. Una volta montati dobbiamo evitare di toglierli nuovamente dal loro posto per non forzare il telaio e correre il rischio che si rompano i punti di ancoraggio.



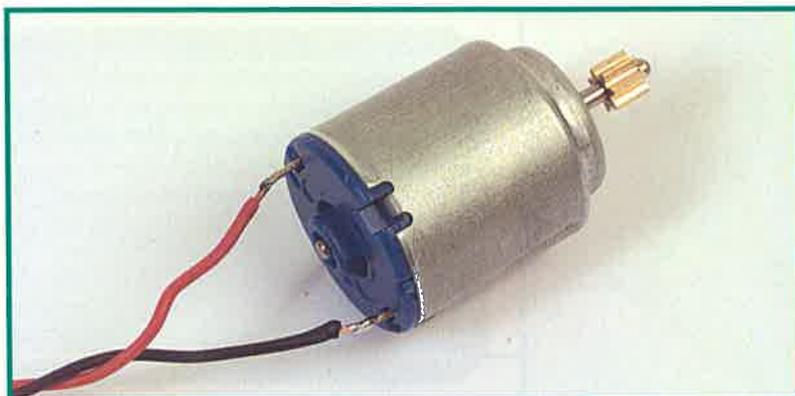
In questa immagine mostriamo come deve essere collegato il coperchio delle pile, che supporta anche la sfera posteriore per lo spostamento, e serve come terzo punto di appoggio del robot. Introduciamo le pile necessarie per il funzionamento del robot e chiuderemo il coperchio, fissandolo bene.

La figura mostra la struttura a forma di torre con tutte le schede montate sul telaio. Le schede sono tenute ferme dalle barre filettate e dalle viti ai loro estremi superiori.



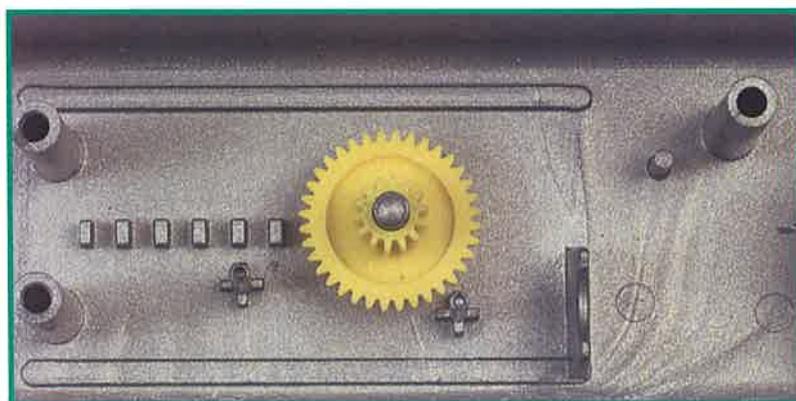
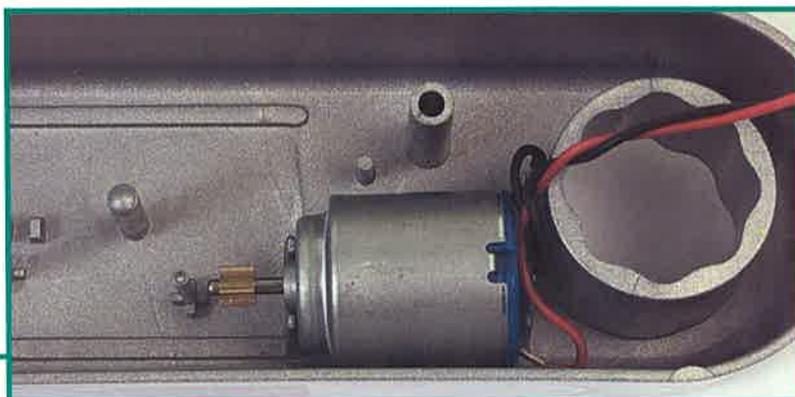
La testa del robot si monta sopra il collo del corpo, che serve anche per bloccarlo e fissarlo bene. La testa inoltre dispone di due clips, che tengono insieme le due metà da cui è formato. Dobbiamo fare pressione su tutti e due gli elementi fino a che non siano ben uniti e fissati.

Montaggio passo a passo



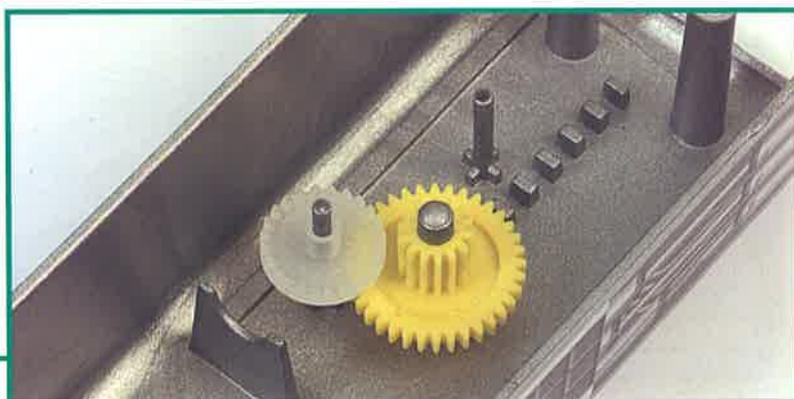
Nell'immagine si vede il motore con i due cavi già saldati ai suoi terminali, pronto per funzionare. I cavi devono essere preparati nel modo abituale. Osservate il pignone presente sull'asse del motore che servirà per azionare il sistema di ingranaggi per il movimento della pinza.

Questa è la posizione nella quale verrà inserito questo motore all'interno del braccio di Monty. I cavi che abbiamo saldato sui terminali andranno al morsetto J2 dei contatti della scheda di controllo della pinza.



Montiamo ora la meccanica necessaria per il movimento della pinza del robot. Il primo pezzo da montare è l'ingranaggio di dimensione intermedia, che va inserito nell'asse come è mostrato nell'immagine. Possiede due corone dentate. Quella più stretta deve rimanere orientata verso l'alto.

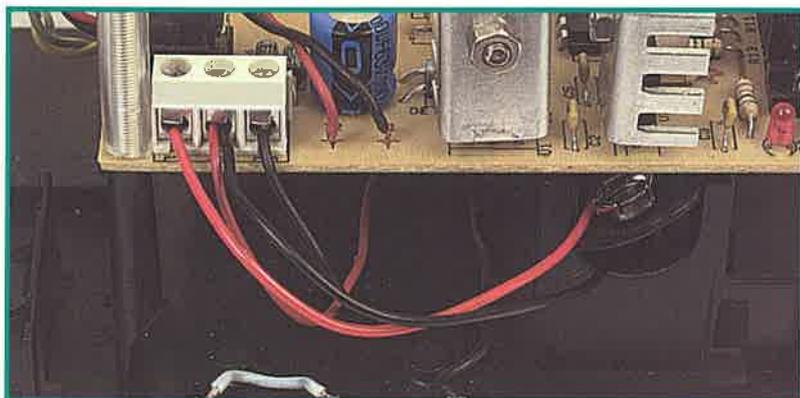
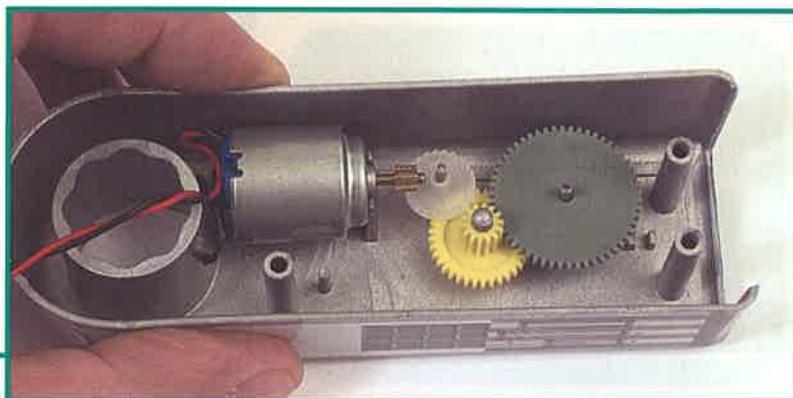
In seguito dobbiamo montare l'ingranaggio più piccolo. La corona di diametro maggiore deve essere orientata verso l'alto. Il suo pignone entra in contatto con l'ingranaggio medio collocato in precedenza.





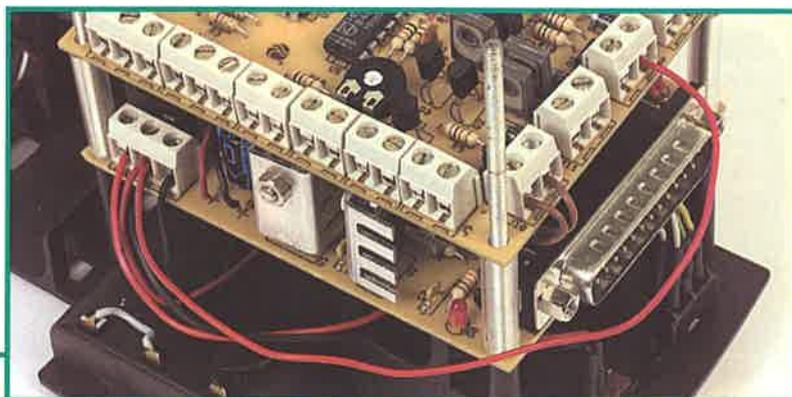
Il terzo ingranaggio da montare è quello più grande. Come si vede nella figura, esso riceve movimento tramite l'ingranaggio medio. Mediante una guida che vi sarà fornita in seguito, l'ingranaggio grande converte il movimento di rotazione in un altro di traslazione.

Infine monteremo il motore, e con esso avremo montato la maggior parte della meccanica necessaria per il movimento della pinza. Una volta montata tutta la struttura mostrata nell'immagine, raccomandiamo di muovere con le dita l'asse del motore per verificare che tutto il sistema di ingranaggi si muova in modo corretto. È conveniente utilizzare un po' di grasso lubrificante che permetta un buon movimento degli ingranaggi.

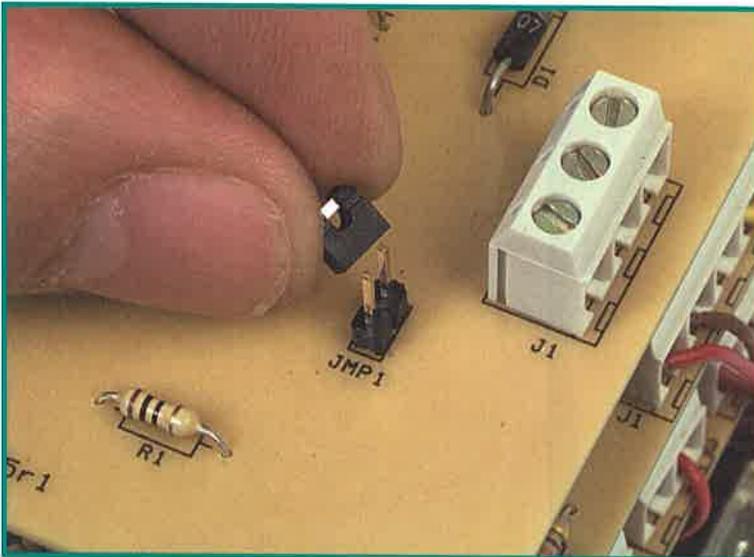


Montiamo ora i cavi di alimentazione tra le pile e la morsettiera J2 della scheda di controllo. Tramite il morsetto 1 (di sinistra) collegheremo il positivo della pila da 9V e sul morsetto 2 (centrale) il negativo della stessa pila. Sullo stesso morsetto 2 collegheremo il positivo del pacco di pile da 1,5V e tramite il morsetto numero 3 (destra) il negativo dell'insieme di pile da 1,5V. In questo modo sia la pila da 9V che le pile da 1,5V rimangono collegate in serie per ottenere i 18V necessari nella scrittura.

Portiamo ora la tensione di alimentazione ai due motori. Questa si estrae dal morsetto 2 (centrale) della morsettiera J2 della scheda di controllo. Mediante il cavo rosso fornito appositamente si collega con il morsetto 2 della morsettiera J13 della scheda di potenza. I motori rimangono alimentati con 9V. La massa è collegata tramite il Bus J1. Il resto delle connessioni dei sensori si realizza in funzione delle applicazioni che desidera l'utente, come è stato spiegato nei vari esempi.



Montaggio finale



La scheda di controllo della pinza ha due modalità di funzionamento, selezionabili tramite il jumper JMP1. Se il jumper è chiuso allora il funzionamento della pinza è temporizzato. Ogni volta che invieremo un fronte di salita tramite la scheda di controllo alla scheda della pinza, quest'ultima si aprirà e trascorso un determinato tempo si chiuderà automaticamente. Se il jumper è aperto, con il primo fronte di salita inviato tramite la linea di controllo faremo aprire la pinza, e con il secondo fronte la faremo richiudere.

Questo primo programma mostra il funzionamento in modo temporizzato, con il jumper chiuso, e supponendo che il segnale di controllo utilizzato per attivare la pinza sia RB4. Il programma consiste nel configurare questa linea di controllo come uscita, per poi attivarla nel momento in cui si desidera aprire la pinza. Trascorso un determinato tempo la pinza si richiuderà automaticamente.

```

c:\progra~1\mplab\assembler2.asm
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
List p=16F84a
include "P16F84a.INC"

org 0x00
goto INIZIO
org 0x05

INIZIO:
    bsf STATUS,RP0
    clrf PORTB
    bcf STATUS,RP0

;Apri la pinza (si chiude automaticamente)
    bsf PORTB,4
    nop
    nop
    nop
    bcf PORTB,4

FINE:
    goto FINE

END
    
```

```

c:\progra~1\mplab\assembler2.asm
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
List p=16F84a
include "P16F84a.INC"

CONTATORE EQU 0x0C

org 0x00
goto INIZIO
org 0x05

DELAY
    movlw .100
    movwf CONTATORE

DELAY10MS
    movlw 0xD8
    movwf TMR0

DEL:
    btfss INTCON,2
    goto DEL
    bcf INTCON,2
    decfsz CONTATORE,1
    goto DELAY10MS

return
    
```

Quest'altro programma mostra come realizzare il controllo della pinza con il jumper aperto. In questo caso dobbiamo inviare due fronti di salita, uno per aprire la pinza e l'altro per chiuderla. In questo esercizio fra l'invio di un fronte e l'altro attendiamo un determinato tempo, stabilito da una routine di configurazione come mostrato nell'immagine.

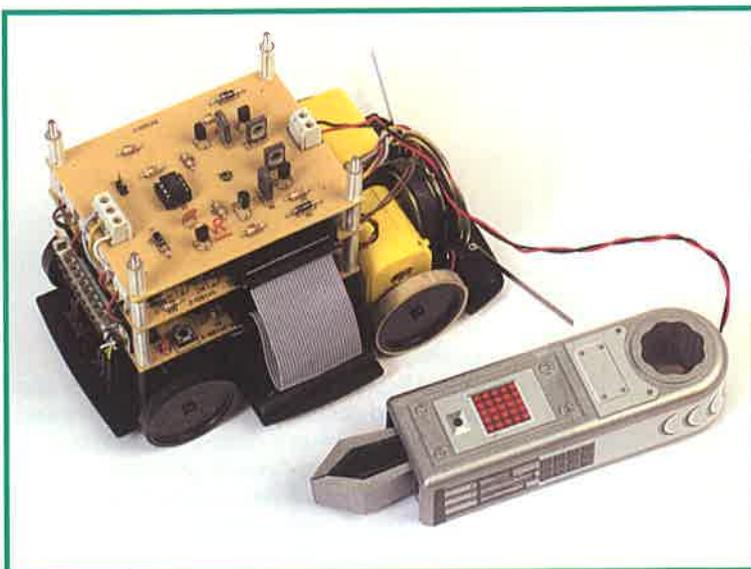
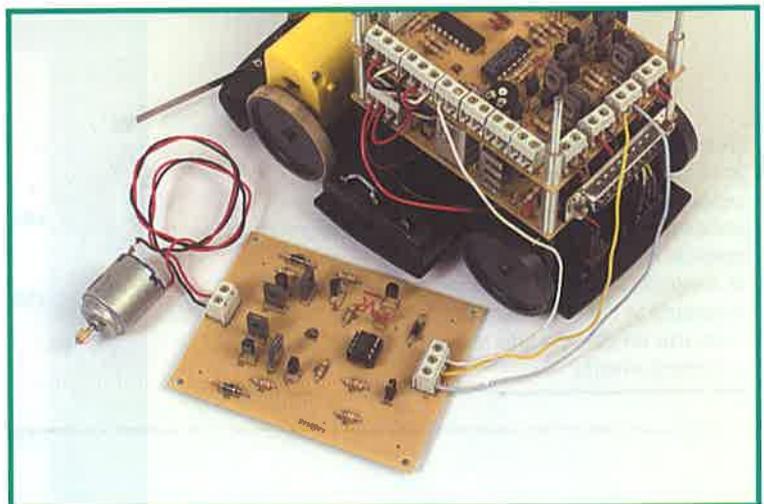
```

C:\prog\1\1\mpio\masc_b0\eseriz.asm
20
21 INIZIO      bsf     STATUS,RP0
22            clr     PORTB
23            movlw  0x07
24            movwf  OPTION_REG
25            bcf     STATUS,RP0
26
27 CICLO      bsf     PORTB,4
28            nop
29            nop
30            bcf     PORTB,4
31            call   DELAY
32            bsf     PORTB,4
33            nop
34            nop
35            bcf     PORTB,4
36            call   DELAY
37            goto   CICLO
38
39            END

```

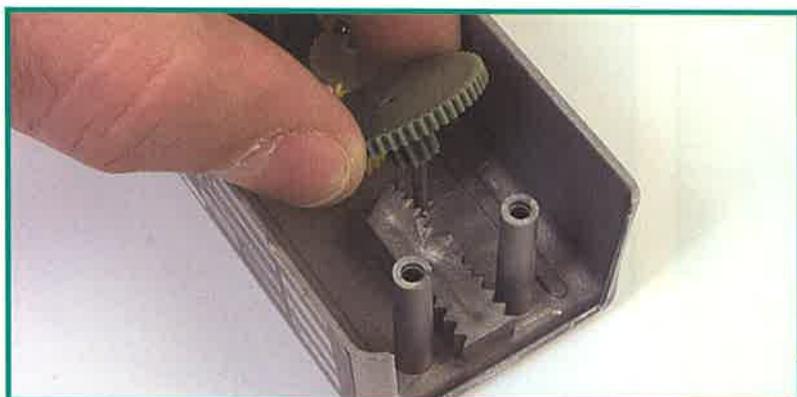
Questa è la parte principale del programma. Consiste in un ciclo che invia due fronti di salita che serviranno per aprire e chiudere la pinza, tramite la linea di controllo RB4. Il programma viene eseguito in modo ciclico, in pratica possiamo gestire l'invio di questi fronti verso la pinza, per raccogliere o posare degli oggetti, in funzione del compito assegnato a Monty in quel momento.

Il motore che controlla l'apertura e la chiusura della pinza si collega tramite la morsettiera J2 della scheda di controllo della pinza. L'alimentazione della scheda e i segnali di controllo si applicano tramite la morsettiera J1. Dobbiamo portare al primo contatto della morsettiera la tensione di alimentazione, che possiamo derivare dalla scheda di potenza. Il secondo contatto è la massa, e il terzo è il segnale di controllo, che per questo esercizio arriva da RB4 (morsetto 6 della scheda di potenza).



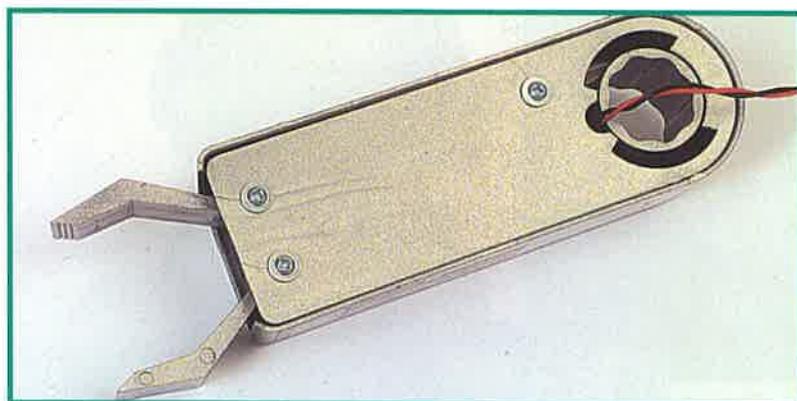
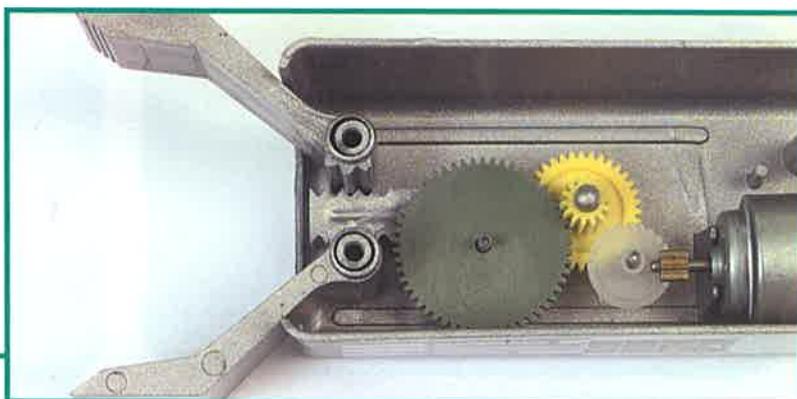
Questo è il montaggio definitivo per provare questo esercizio. Dobbiamo collegare la scheda di potenza tramite la scheda di controllo, e la scheda di controllo della pinza tramite la scheda di potenza, come abbiamo appena spiegato. Il motore lo possiamo inserire all'interno della struttura del braccio, per poter verificare il reale funzionamento di apertura e di chiusura della pinza, a seconda di come inviamo i segnali di controllo.

Montaggio passo a passo



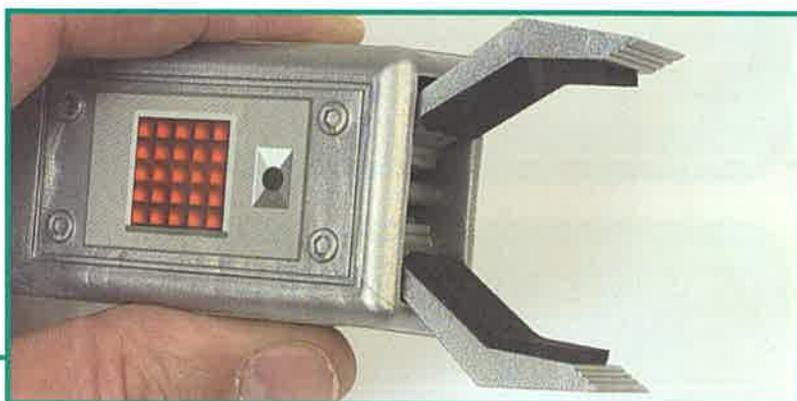
Per terminare il montaggio della struttura della pinza del robot dobbiamo montare la guida del braccio nella posizione mostrata dall'immagine, facendo in modo che si incastrino con l'ingranaggio grande, posizionato all'interno del braccio.

Ora monteremo la pinza del braccio, che è composta da due semipinze. Inseriremo le due metà nei supporti presenti sull'estremo del braccio. Dobbiamo fare in modo che la parte dentata delle semipinze ingrani con la guida dentata del braccio.



Dopo aver fissato la meccanica di controllo della pinza, monteremo il coperchio del braccio. Dobbiamo montare il coperchio per il verso giusto, come mostrato dall'immagine, e fare una leggera pressione sino a farlo incastrare correttamente. Avviteremo il coperchio senza forzare, per evitare blocchi nel meccanismo. L'altro coperchio servirà per chiudere il secondo braccio di Monty; entrambe le braccia ora sono pronte per essere montate sul corpo di Monty.

Come ultima operazione taglieremo il pezzo di gommapiuma adesiva in due parti uguali, che monteremo all'interno della pinza del robot. In questo modo sarà più facile afferrare gli oggetti per trasportarli dove vorremo.

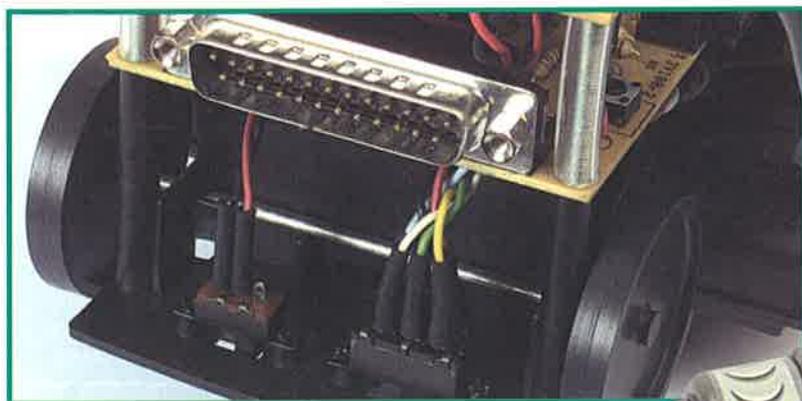




Nell'immagine possiamo vedere il montaggio del braccio della pinza sul corpo di Monty; si tratta del braccio destro del robot. I cavi del motore si inseriscono all'interno della spalla per poterli collegare alla morsettiera J2 della scheda di controllo del braccio. Il braccio si fissa mediante la corrispondente clip.

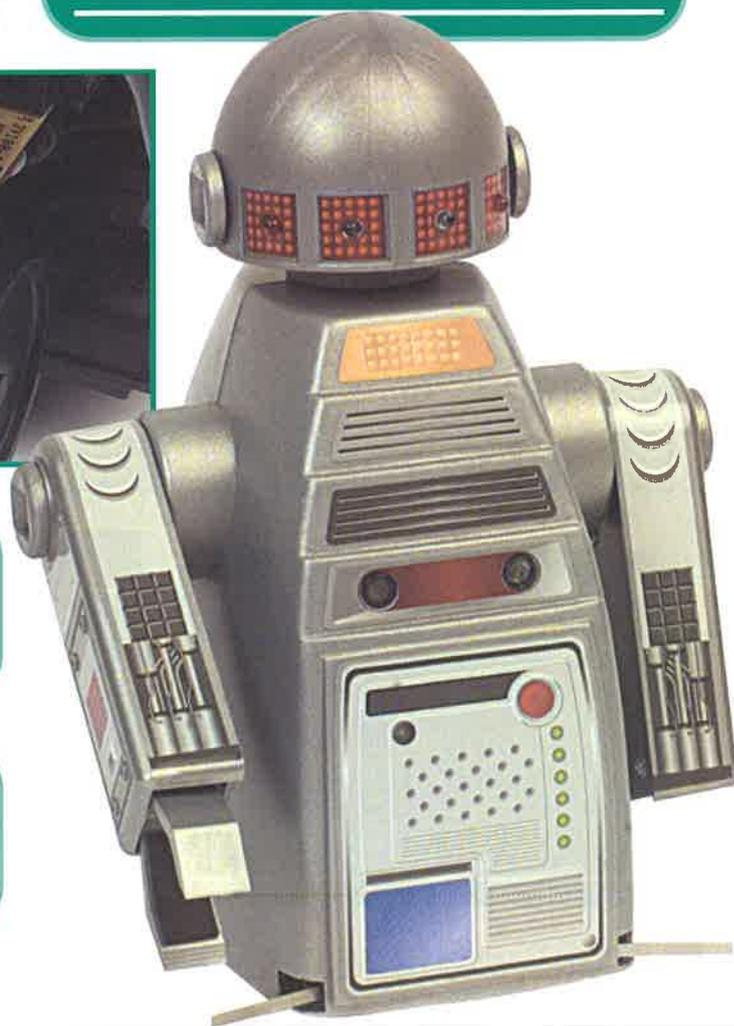


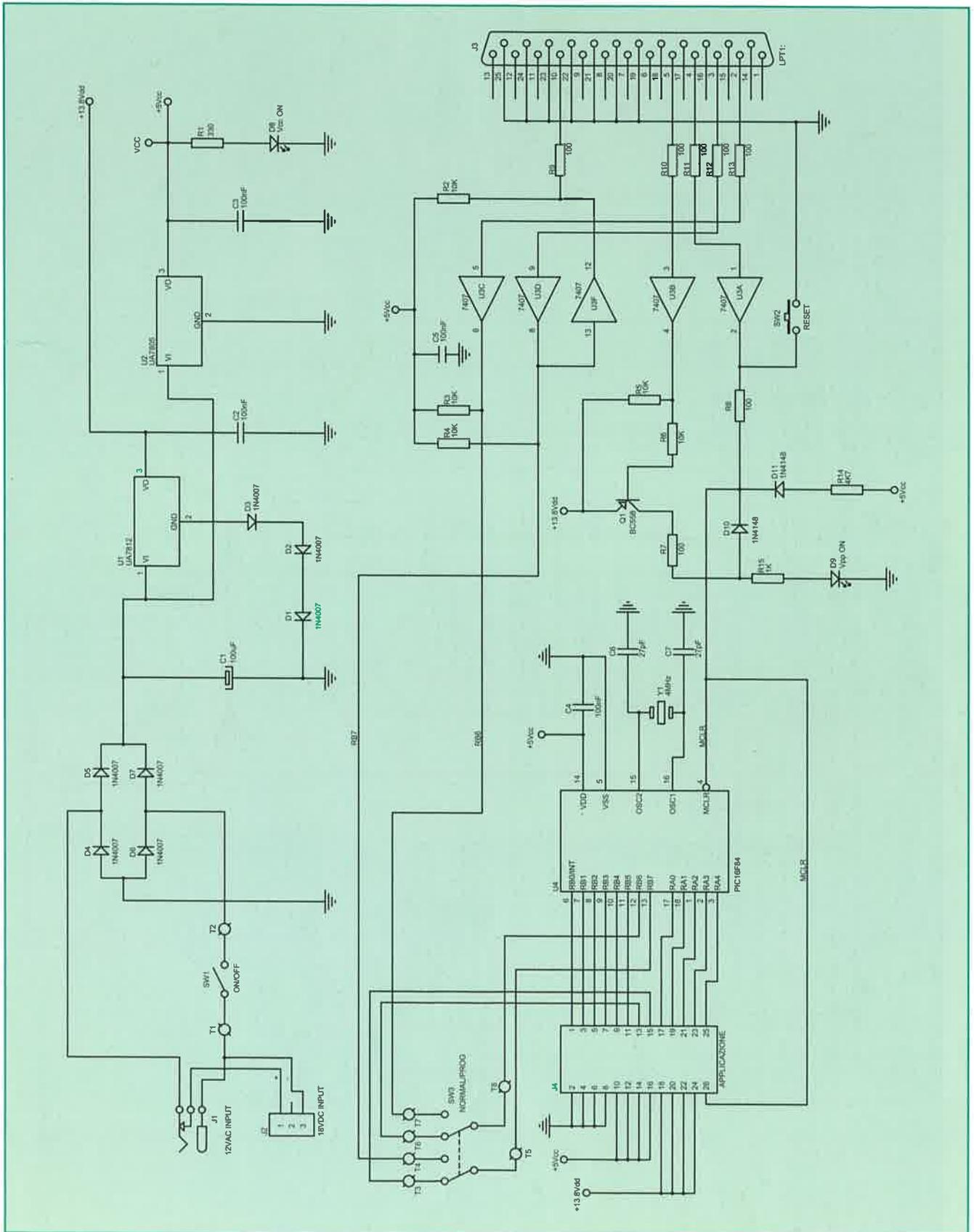
Seguendo una procedura simile si monta il braccio sinistro di Monty. Anche quest'ultimo si fissa con la corrispondente clip.



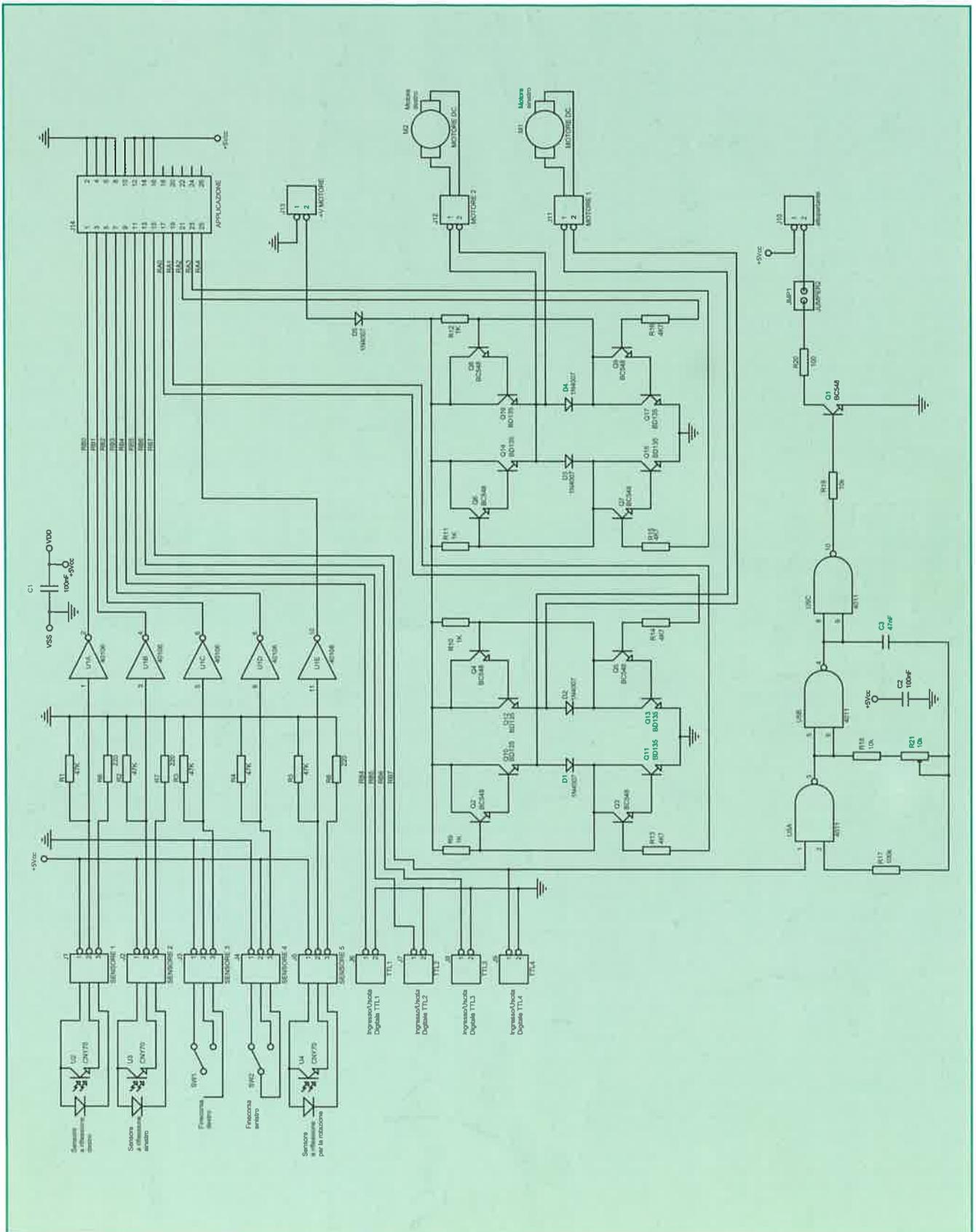
Sono state fornite due ruote e un alberino; nell'immagine vediamo come fissare entrambe le ruote e l'alberino nella parte posteriore di Monty. Queste ruote sono solo decorative e non influiscono sui movimenti.

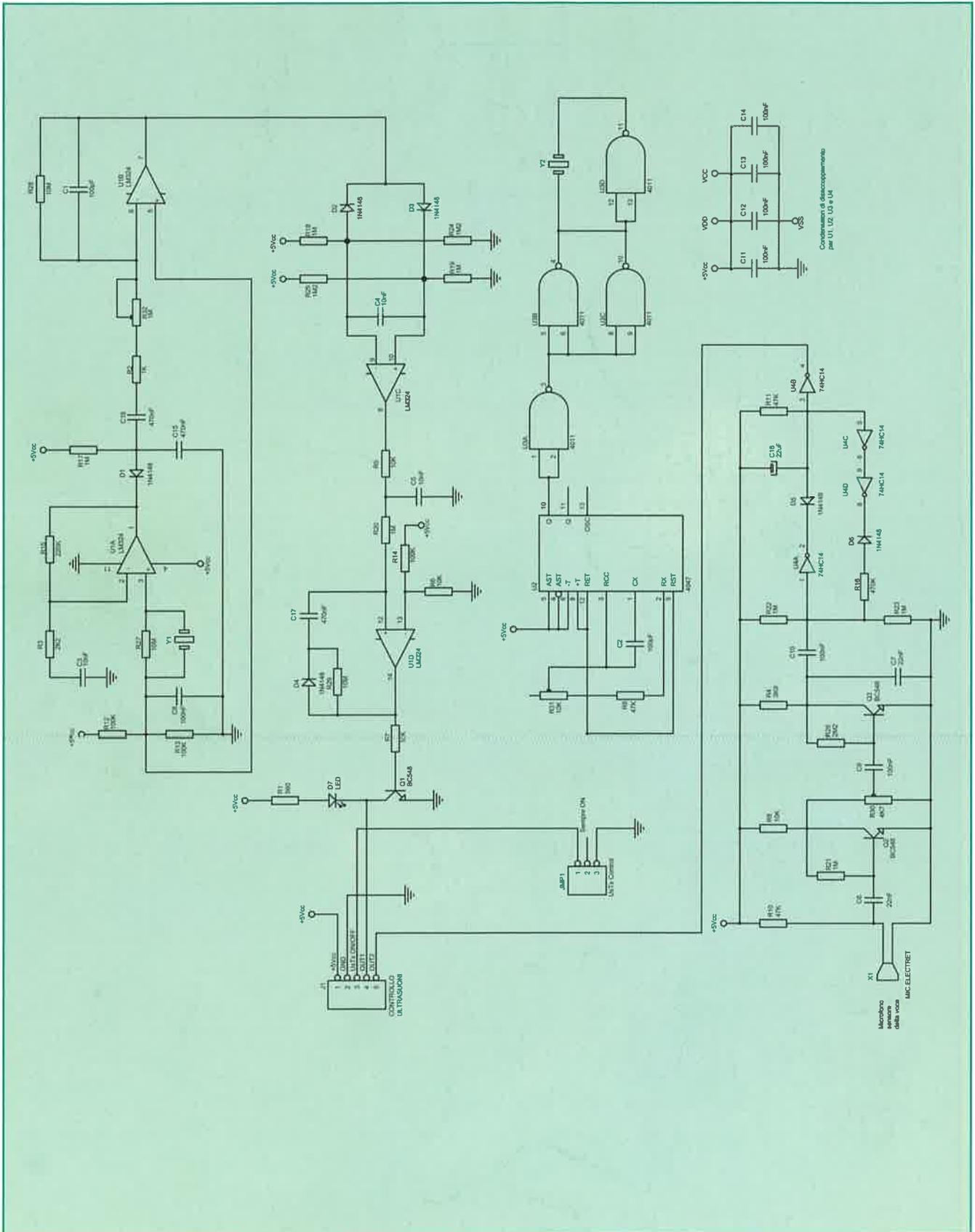
Nell'immagine vediamo Monty completamente terminato, e pronto per funzionare.

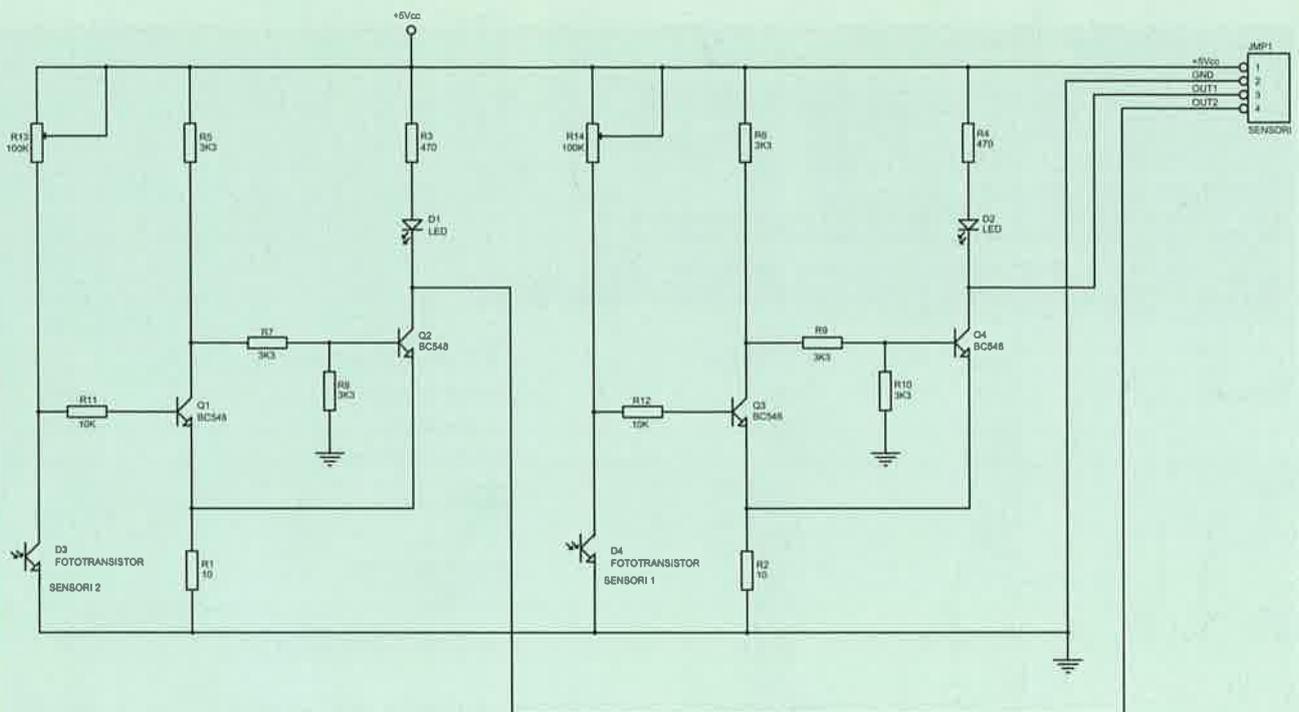




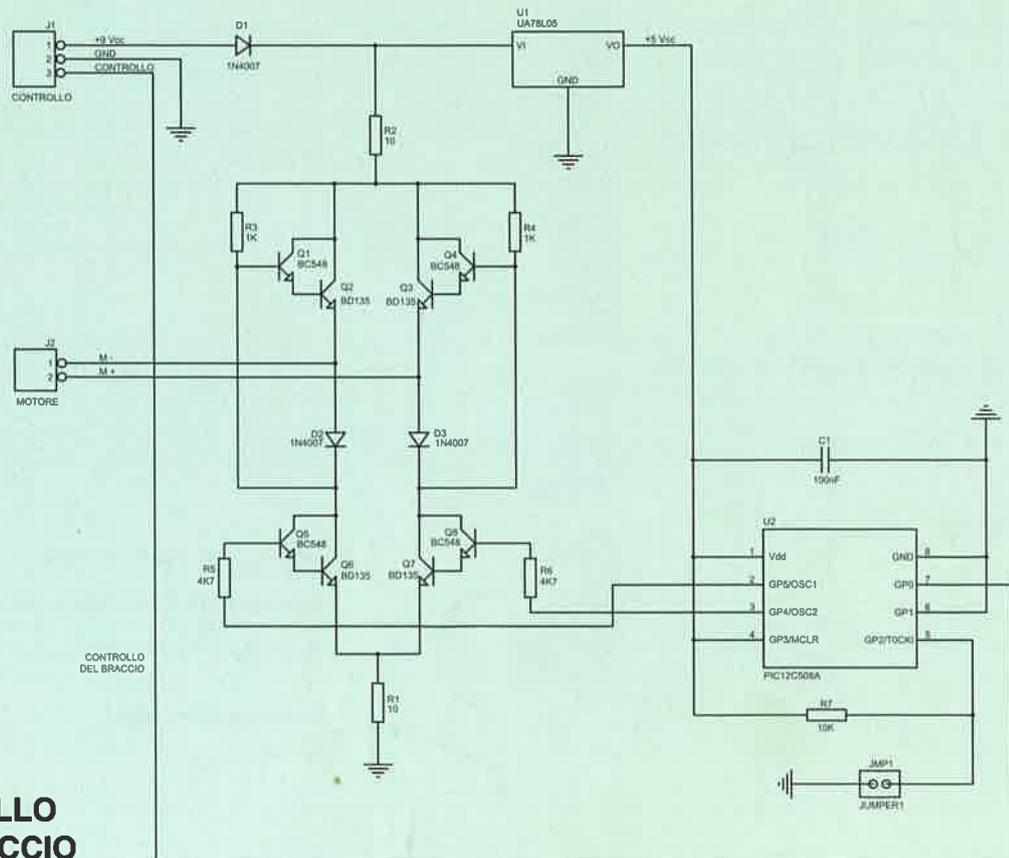
Modulo di potenza







MODULO DELLE LUCI



CONTROLLO DEL BRACCIO

In questa applicazione combiniamo i diversi sensori di Monty per fare in modo che il robot possa reagire a seconda degli stimoli dell'ambiente, svolgendo la funzione di poliziotto, il robot inizia da fermo, in attesa di un suono, ad esempio quello di un battere di mani per muoversi. Quando rileva un suono, il robot inizia a muoversi aprendo e chiudendo la pinza in modo temporizzato, con il jumper JMP1 del controllo del braccio chiuso. Se i finecorsa rilevano un urto contro qualche cosa, il robot retrocede, dopodichè riprende la marcia. Se mentre sta pattugliando si fa notte (assenza di luce) il robot si ferma, fino a che non rileva un movimento. Quando rileva il passaggio di un oggetto mobile dinnanzi a se, apre e chiude la pinza. Estrarre il jumper dell'altoparlante JMP1 dalla scheda di potenza.
 RB2 e RB3 finecorsa meccanici -> Collegamento sulle morsettiere J3 e J4 della scheda di potenza
 RB6 sensore di suono -> Collegamento sulla morsettiere J8 (1) della scheda di potenza dalla scheda dei sensori J1 (5)
 RB7 pinza del braccio -> Collegamento sulla morsettiere J9 (1) della scheda del braccio J1 (3)
 RB4 sensore della luce -> Collegamento nella morsettiere J6 (1) della scheda di potenza dalla scheda delle luci J1 (3)
 RB5 ultrasuoni -> Collegamenti nella morsettiere J7 (1) della scheda di potenza tramite la scheda dei sensori J1 (4)
 Motori collegati alla morsettiere J12 e J11 della scheda di potenza.

```

List include          D=16F84a
                    "P16F84a.INC"
; Tipo di processore
; Definizione dei registri interni

CONTATORE           EQU          0x0C
                    ; Variabile ausiliaria

ORG                 0x00
goto                INIZIO
ORG                 0x05
                    ; Vector di Reset
                    ; Salva il vector di interrupt

INIZIO              bsf          STATUS,RPO
                    drf          PORTA
                    movlw       b'01111100'
                    movwf      PORTB
                    movlw       b'00000111'
                    movwf      TMR0
                    bcf          STATUS,RPO
                    drf          PORTA
                    ; Banco 1
                    ; Pin di ingresso e uscita

START:              call        DELAY1S
                    btss       PORTB,6
                    goto        START
                    btss       PORTB,6
                    goto        START
                    movlw       b'00000110'
                    movwf      PORTA
                    ; Temporrizzazione di 1 secondo
                    ; attende che si produca un suono

; Ciclo principale del programma, che verifica lo stato dei vari
; sensori per agire di conseguenza.
PATTUGLIA:         btfs       PORTB,2
                    goto        URTO
                    btfs       PORTB,3
                    goto        URTO
                    bsf         PORTB,7
                    call        DELAY1S
                    bcf         PORTB,7
                    btfs       PORTB,4
                    goto        SENZA_LUCE
                    goto        PATTUGLIA
                    ; Apre e chiude la pinza
                    ; Temporrizzazione di 1 secondo

; Se il robot urta, fa retromarcia aprendo e chiudendo la pinza
URTO:              movlw       b'00001001'
                    movwf      PORTA
                    bsf         PORTB,7
                    call        DELAY1S
                    bcf         PORTB,7
                    call        DELAY1S
                    bsf         PORTB,7
                    call        DELAY1S
                    bcf         PORTB,7
                    movlw       b'00000110'
                    movwf      PORTA
                    goto        PATTUGLIA
                    ; Retromarcia
                    ; Apre e chiude la pinza

; Se il robot rimane al buio si ferma in attesa di un movimento
SENZA_LUCE:       drf          PORTA
                    btfs       PORTB,5
                    goto        SENZA_LUCE
                    ; Testa gli ultrasuoni

FINALE:           bsf         PORTB,7
                    call        DELAY1S
                    bcf         PORTB,7
                    goto        SENZA_LUCE
                    ; Pinza ON

; Routine che realizza un ritardo di 1 secondo
DELAY1S:         movlw       100
                    movwf      CONTATORE
                    movlw       0xD8
                    movwf      TMR0
                    intcon,2
                    del 10
                    intcon,2
                    movlw       CONTATORE,F
                    decfsz     CONTATORE,F
                    goto        DELAY10MS
                    ; Inizializza la variabile Contatore con 100.
                    ; Carica il Timer0
                    ; Attende che il Timer0 vada in overflow (10 ms)

DELAY10MS       movlw       100
                    movwf      DEL10
                    btss       PORTB,7
                    goto        DEL10
                    bcf         DEL10
                    decfsz     DEL10
                    goto        DELAY10MS
                    ; Ripete la routine 100 volte (100x10ms = 1s).

END              ; Fine del programma sorgente
    
```