



# Contatori

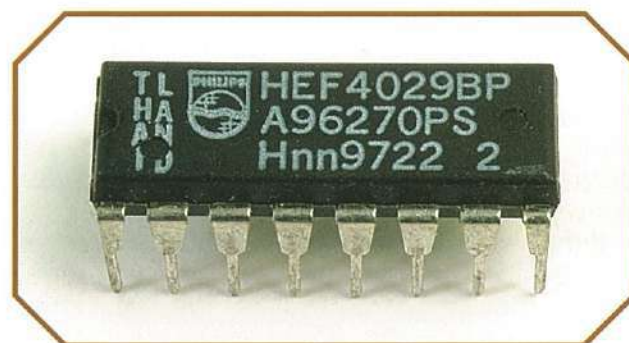
**N**ell'elettronica digitale il termine contatore si applica in modo specifico ai circuiti con un segnale di uscita di diversi bit che rappresentano un numero in binario e che, ricevendo un segnale di clock di ingresso, avanzano o retrocedono di una posizione all'interno di una tabella, che normalmente segue l'ordine numerico naturale. Questa definizione può spaventare, per questo vi spiegheremo in modo più ampio il suo significato.

## Contatori

Sono i dispositivi che ci indicano il turno di attesa negli uffici amministrativi o nei supermercati, i segnapunti negli stadi di calcio, il contatore di pezzi delle macchine industriali o quelli utilizzati nelle fotocopiatrici per il conteggio dei fogli, oppure negli orologi digitali che ci informano del giorno e dell'ora in cui viviamo.

## Driver

Tutti i dispositivi che abbiamo menzionato prima sono basati su contatori elettronici che utilizzano un codice binario, però si leggono in decimale. Quindi, è necessario un circuito



Circuito integrato contatore da quattro bit.

che in qualche modo trasformi il codice binario in un segnale adatto al dispositivo di presentazione utilizzato.

Nel caso del nostro laboratorio si utilizza un display LED a 7 segmenti, quindi avremo bisogno di un circuito che trasformi il codice BCD in 7 segmenti, ovvero avremo quattro ingressi corrispondenti ai quattro bit che sono necessari per rappresentare i numeri dallo 0 al 9, e sette uscite corrispondenti ad ogni segmento.

Questo tipo di circuito prende il nome di DRIVER.

## Numero di bit

Il numero di bit di un contatore dipende dal numero massimo che si vuole rappresentare: con tre bit si rappresentano solo numeri dallo 0 al 7, risultando insufficiente per i display che necessitano di rappresentare numeri dallo 0 al 9. Quindi aggiungeremo un bit in più, in modo da poter rappresentare 16 valori, da 0 a 15, anche se ne servono solo 10, da 0 a 9.

Normalmente i circuiti integrati contatori come il 4090 hanno la possibilità di selezionare il modo di conteggio per 16 valori (BINARIO) o 10 valori (DECADE), a questo scopo possiedono un terminale indicato come B/D.



Schema a blocchi di base di un contatore.

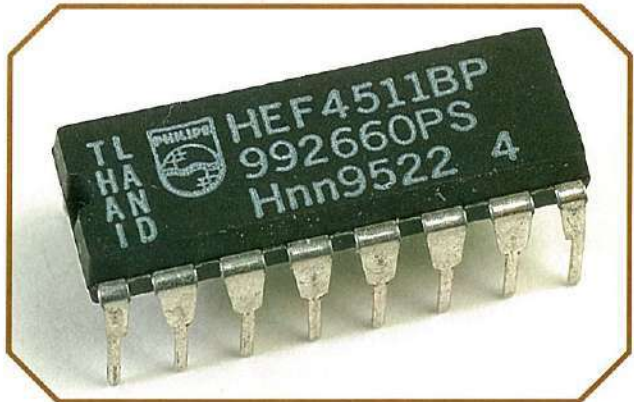


## Segnale di clock

Anche se molti lettori avranno già delle conoscenze in merito, altri che sono agli inizi della navigazione nell'appassionante mondo dell'elettronica digitale, possono confondersi per colpa della terminologia.

Non è difficile capire che la complicata circuiteria di un computer, oppure di un dispositivo semplice come un contatore, deve lavorare in modo che il suo ritmo di funzionamento sia supportato anche dagli altri componenti cui è collegato, ovvero che riceva ogni segnale nel momento opportuno, né prima, né dopo. In altre parole, i circuiti devono lavorare in modo sincronizzato.

Per stabilire questa "sincronizzazione" è necessario disporre di un oscillatore che segni il tempo di lavoro: questo oscillatore si chia-



Circuito integrato driver da quattro bit di ingresso con uscita per il display a LED a 7 segmenti.

ma clock. In dispositivi complessi possiamo trovare diversi clock a frequenze differenti.

È evidente che quanto più veloce è il clock, tanto più veloce sarà la macchina, di conseguenza è la velocità massima cui possono lavorare i circuiti integrati. Attualmente in alcuni microprocessori si superano già i 2 GHz.

## Esempio

Per costruire un clock che misuri il tempo è sufficiente un oscillatore che generi con precisione una frequenza di clock di 1 secondo. Utilizzando diversi contatori si possono contare i secondi da 0 a 59, in modo che ogni volta che si supera il 59 tornino a zero e avanzi di una unità il conteggio dei minuti, e quando quest'ultimo raggiunge il 59 ritorni a zero avanzando di una unità il conteggio delle ore.

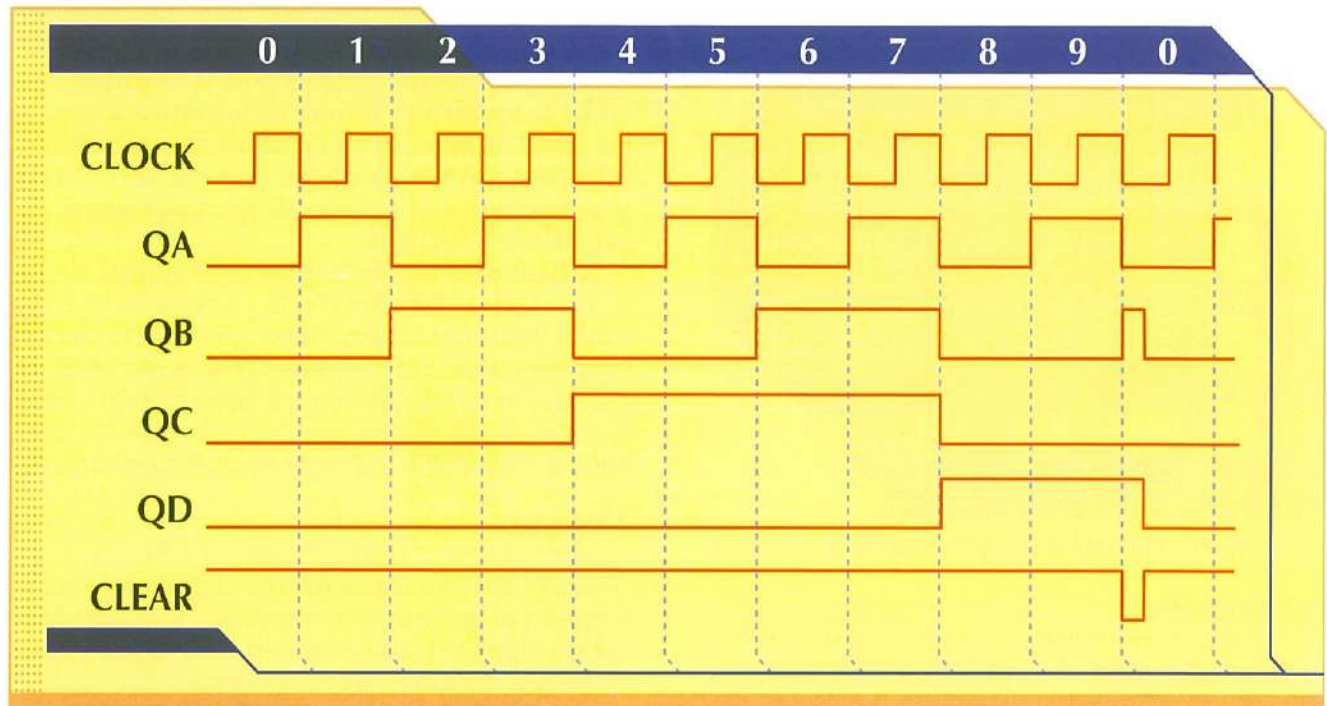
Costruire un orologio che lavori a 24 o 12 ore è realmente semplice, basta passare da 11:59 a 00:00, o da 23:59 a 00:00. Tutto questo verrà spiegato in modo più dettagliato.

## Cronogrammi

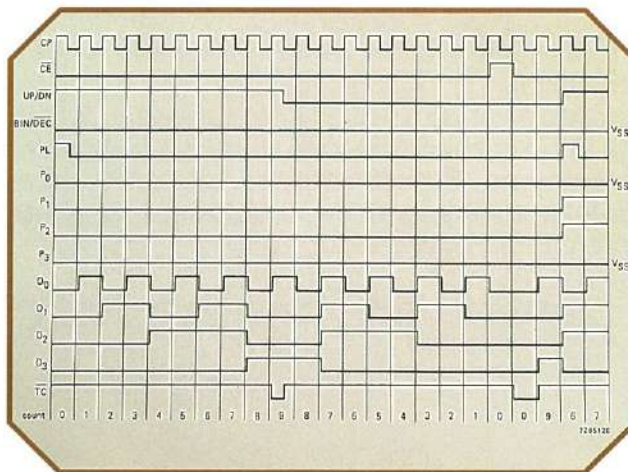
Quando si lavora con circuiti i cui segnali variano con il tempo è necessario utilizzare i cronogrammi. Nei cronogrammi di base si può vedere l'evoluzione dell'uscita in funzione degli ingressi e la loro evoluzione nel tempo. Esiste una linea di cronogramma per ogni segnale. La frequenza di funzionamento viene indicata come CK, che deriva dalla parola anglosassone Clock.

	BINARIO			DECIMALE
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Equivalenza tra codice binario e decimale.



Cronogramma di base di un contatore a quattro bit.



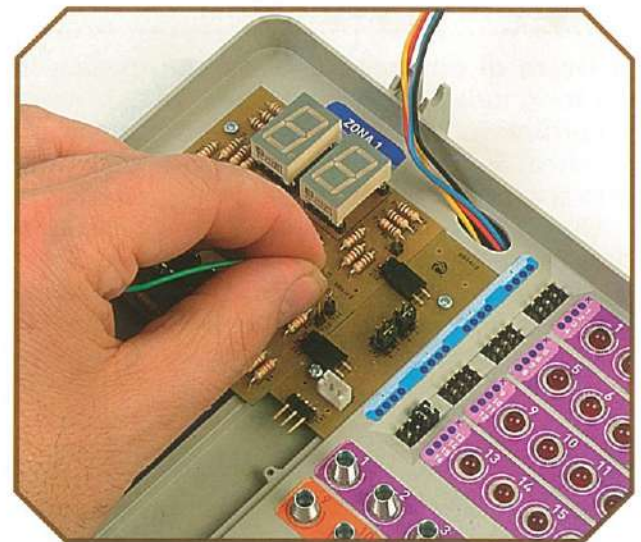
Cronogramma di un 4029 commerciale a quattro bit.

Per ogni segnale, che è sotto forma di impulso, il livello alto di tensione si attribuisce all'uno logico e quello basso allo zero logico. Le uscite dei contatori vengono abitualmente chiamate QA, QB, QC e QD, essendo QA il bit meno significativo e QD il più significativo o di maggior peso; è anche utilizzata la denominazione numerica Q1, Q2, Q3 e Q4.

Osservando il cronogramma si può vedere l'avanzamento del conteggio.

Nei cronogrammi più completi si combinano più segnali, specialmente quelli di controllo.

Su Internet è facile trovare i "Data sheet" di questo tipo di circuito e osservare i loro cronogrammi, che, normalmente, si possono scaricare in formato pdf.



Circuito corrispondente al display e al driver del laboratorio.

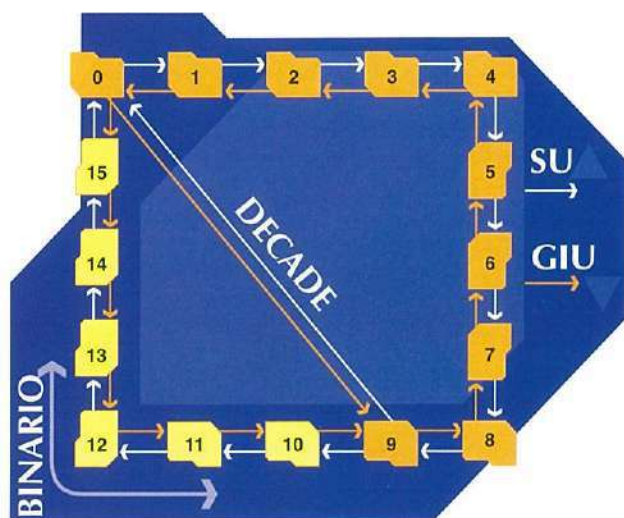


Diagramma degli stati di un contatore 4029.

## UP/DOWN

Ci sono contatori che realizzano il conteggio solo nel senso ascendente (UP), ovvero quando ricevono l'impulso di clock incrementano il conteggio. Altri, al contrario, lo realizzano in senso inverso, ossia fanno il tipico conto alla rovescia (DOWN). Esistono anche contatori reversibili che possono realizzare il conteggio in un verso o nell'altro, ad esempio, il circuito integrato 4029 ha un terminale, il 10, che quando è a zero conta in senso discendente e quando è a uno lo fa in senso ascendente.

## Diagramma degli stati

Si tratta di una rappresentazione grafica in cui sono indicati i possibili stati del contatore e l'ordine in cui si può passare da uno stato all'altro. Se c'è un salto che non si può fare, non si rappresenta la linea corrispondente.

Questi diagrammi sono molto interessanti perché già a prima vista si può capire la sequenza ascendente o discendente che può seguire un contatore, e anche se non sono riportati i livelli dei segnali risultano molto intuitivi.

## L'interno

Esistono molti tipi di contatori, però quello che ha il funzionamento più elementare è il contatore sincrono.

Per un contatore da quattro bit abbiamo

bisogno di quattro bistabili, che spiegheremo nel dettaglio più avanti.

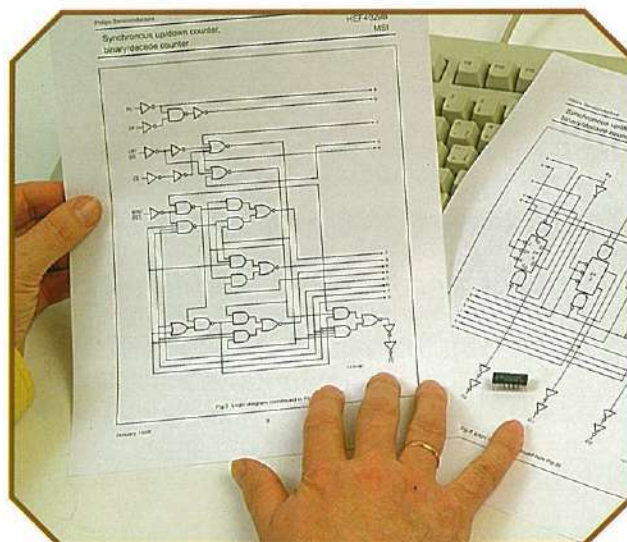
In un contatore sincrono l'ingresso del clock si applica in modo simultaneo a ciascuno degli ingressi di clock dei quattro bistabili. In questo caso la velocità massima del conteggio dipende dal tempo di propagazione del segnale nei bistabili.

Tratteremo nel dettaglio tutti questi circuiti allo scopo di facilitare la comprensione del loro funzionamento, però, per il momento, occorre avere molto chiaro che le uscite di un contatore cambiano con il tempo e in modo sincronizzato al segnale di clock, che fa avanzare il conteggio ad ogni impulso generato.

## Preset

Alcuni contatori necessitano di iniziare il conteggio da un numero specifico, il quale deve essere presente su determinati ingressi prima di essere caricato. Questa operazione si chiama carico in parallelo, che avviene simultaneamente su tutti i bit.

Il circuito integrato 4029 è un contatore sincrono, e si può configurare come ascendente o discendente e come contatore binario a quattro bit (0-15) o decadico (0-9), che molti costruttori hanno a listino e nei cataloghi. Si tratta di un circuito MSI, cioè di media scala di integrazione. È realmente un contatore abbastanza completo e verrà utilizzato in alcuni esercizi del nostro laboratorio.



Circuito interno di un 4029.



# Multiplexer

**I multiplexer è un dispositivo che seleziona l'ingresso dei dati fra più ingressi a disposizione e trasmette l'informazione di questo ingresso ad un'unica uscita. Il principio fondamentale di un multiplexer è quello di un commutatore, però questo commutatore ha un controllo digitale. Si tratta di circuiti molto utilizzati, singolarmente o all'interno di dispositivi più complessi.**

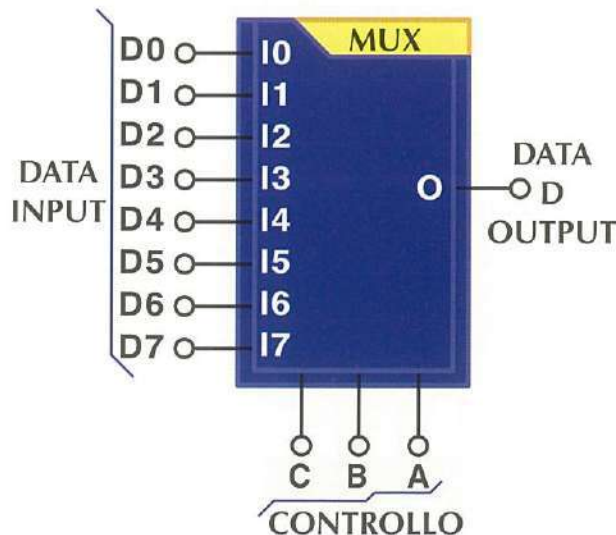
## Numero di ingressi

Il numero di ingressi di un multiplexer è determinato da quello dei terminali di controllo, ovvero, dal numero di bit utilizzato per controllare questi ingressi. In questo modo, con un ingresso di controllo, avremo due ingressi di dati; con due ingressi di controllo, quattro ingressi di dati; con tre ingressi di controllo, otto ingressi di dati; con quattro ne avremo sedici, e così via. Come si può vedere, per ogni bit o terminale di controllo che si aggiunge, si duplica il numero di ingressi da multiplexare.

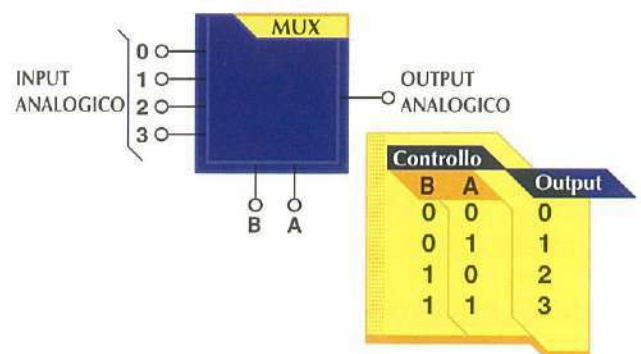
## Multiplexer analogici

L'utilizzo dei multiplexer analogici è molto diffuso, in questo caso i segnali d'ingresso e di uscita sono analogici, mentre quelli di controllo sono digitali.

Un impiego molto semplice per questo tipo di multiplexer è la commutazione audio, applicata ad esempio ad un preamplificatore



Multiplexer di dati a otto ingressi.



Multiplexer analogico a quattro ingressi.

per "selezionare" un ingresso fra i diversi possibili e connetterlo ad un amplificatore di potenza: in questo caso l'applicazione del circuito è praticamente diretta.

## Multiplexer digitali

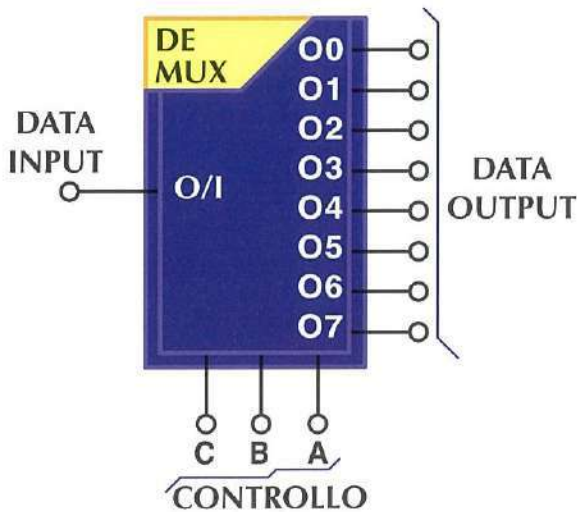
In questo tipo di circuiti i segnali che si attribuiscono agli ingressi sono dei dati, l'uscita è un'uscita di dati e il controllo continua a essere digitale.

Per ottenere un dato in uscita bisogna applicare ai terminali di controllo il codice binario corrispondente al numero dell'ingresso. Ad esempio, per un multiplexer da 8 a 1, se vogliamo portare il dato della linea 5 sull'uscita, sarà necessario attribuire ai terminali di controllo il codice 101, ovvero 5 espresso in binario.

Una delle applicazioni per questo tipo di multiplexer è la trasmissione dei dati.

## Demultiplexer

Il demultiplexer compie la funzione inversa del multiplexer, ha una linea di ingresso e i dati che riceve tramite questa linea li distribuisce alle sue uscite a seconda del codice binario applicato ai terminali di controllo.



Demultiplexer a otto uscite.

Come potrete osservare, si utilizza il termine "linea di ingresso": questo termine non è utilizzato a caso né si tratta di un capriccio linguistico, perché il segnale viene ricevuto tramite un filo, si applica al demultiplexer e questo lo distribuisce alle diverse linee di uscita.

### Multiplexer-demultiplexer

Abbiamo già visto i concetti fondamentali di un multiplexer e di un demultiplexer, ora supponiamo di avere i segnali di un bus a 8 bit, però, prima di proseguire, spieghiamo brevemente ai meno esperti che per bus si intendono gli 8 bit in parallelo collegati agli otto fili alle cui estremità è possibile, tramite un multimetro, misurare se il dato è uno

oppure zero. Se applichiamo questi otto fili agli otto ingressi di un multiplexer, avremo sull'uscita, ordinati secondo un determinato criterio, i dati di ognuno degli ingressi.

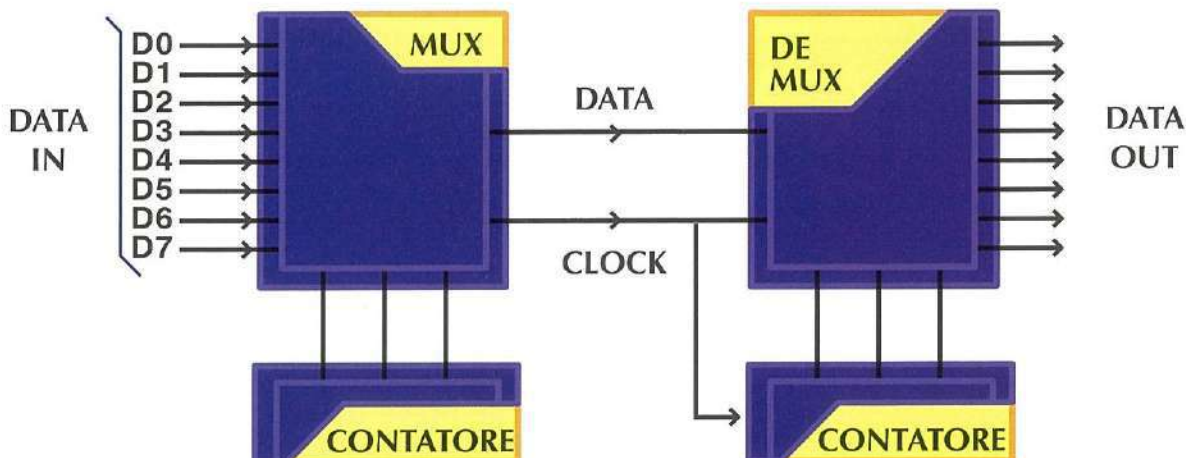
Per fare in modo che i dati siano portati sull'uscita secondo un certo ordine, possiamo collegare un contatore sui terminali di controllo, il quale avanzerà grazie ad un generatore di impulsi, ovvero, ad un clock.

In altre parole, con questa operazione ridurremo gli otto fili, o linee di dati, ad uno solo. Questa linea di uscita la porteremo ad un demultiplexer piuttosto lontano.

Se realizziamo il processo contrario, applicando un contatore ai terminali di controllo del demultiplexer, otterremo sulle uscite i dati che prima avevamo sugli otto fili, e che ora possiamo nuovamente trasmettere su otto linee, recuperando così il segnale originale.

### Trasmissione dei dati

Nel paragrafo precedente abbiamo descritto un rudimentale sistema che riceve i dati in parallelo, li trasmette su di una sola linea – uno dopo l'altro – cioè in serie e successivamente torna a recuperarli in parallelo. Qui si può vedere, anche se in forma molto rudimentale, la conversione dei dati da parallelo a seriale e da seriale a parallelo, oltre alla trasmissione dei dati in seriale su di una sola linea. Questo circuito è molto semplice, però miglioreremo il modello prendendo in considerazione altri concetti.



Principio di base della trasmissione di dati seriali.



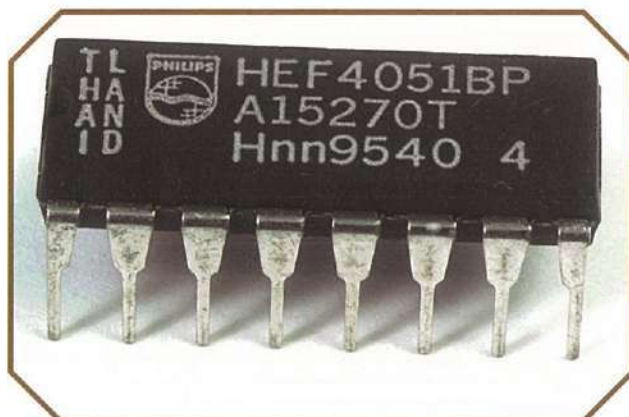
Abbiamo parlato di un contatore che, durante il suo conteggio, originava una determinata sequenza di bit di controllo per il multiplexer, e di un altro contatore che faceva la stessa cosa per il demultiplexer; uno dei punti cruciali di quest'ultimo circuito è che il codice binario che si applica al demultiplexer deve corrispondere al dato che arriva in quell'istante per fare in modo che l'uscita sia adeguata. In altri termini, i contatori devono essere sincronizzati e uno dei modi più semplici per farlo è quello di portare il segnale di clock su un'altra linea. Per quanto questa soluzione sia semplice, il circuito può già funzionare in modo pratico, anche se con molte limitazioni. Introduciamo, quindi, il concetto di sincronizzazione. In un sistema reale di trasmissione dei dati, vengono utilizzati circuiti e protocolli per poter ottenere i segnali che entrano nel sistema. Con il metodo del multiplexer si riesce a trasmettere un alto numero di canali di dati con un ridotto numero di fili.

### Circuiti commerciali

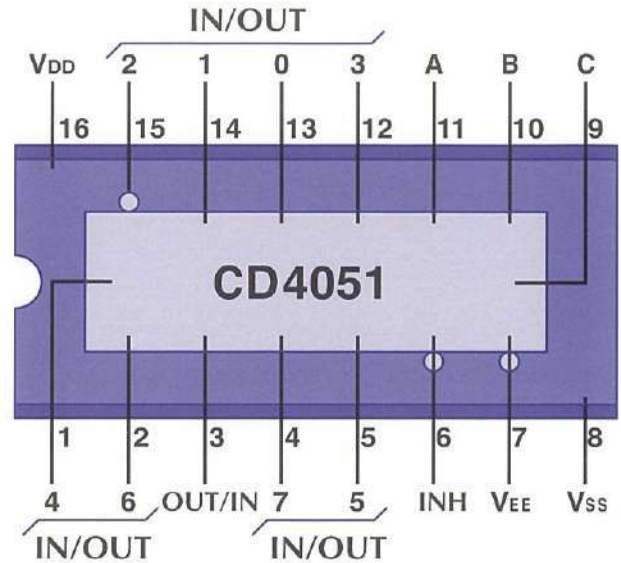
Esistono molte famiglie di circuiti logici che comprendono i multiplexer. Molti modelli sono circuiti combinatoriali, ovvero, sono formati da porte logiche e il loro utilizzo semplifica molto il progetto del circuito, dato che diversamente bisognerebbe costruirli utilizzando grandi quantità di porte logiche.

### CI 4051

Questo circuito integrato di basso consumo CMOS è molto utilizzato nonché facile da tro-



Circuito integrato 4051.



Multiplexer analogico 4051.

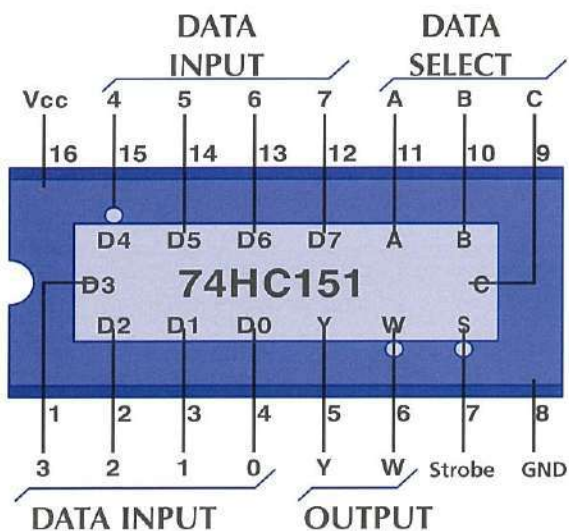
vare nei negozi. È di tipo analogico e si può utilizzare anche per i dati, facendo però attenzione a mantenere ai bit di controllo il tempo sufficiente affinché il dato passi correttamente all'uscita.

In generale i circuiti CMOS sono piuttosto lenti, però hanno costi e consumi molto bassi. Il loro utilizzo è conveniente quando non si necessita di una grande velocità di esecuzione.

Questo multiplexer ha otto ingressi, un'uscita e tre ingressi di controllo, oltre ai due terminali di alimentazione, cosa che non va dimenticata dal momento che frequentemente ci si concentra solo sui collegamenti dei segnali digitali, dimenticando di collegare l'alimentazione.



Circuito integrato 74HC151.



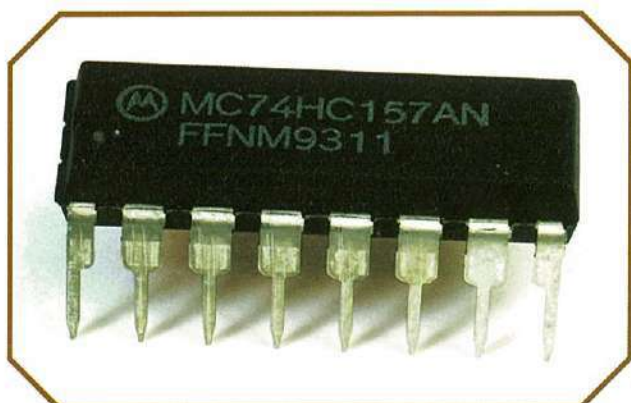
Multiplexer digitale 74HC151.

## CI 74151

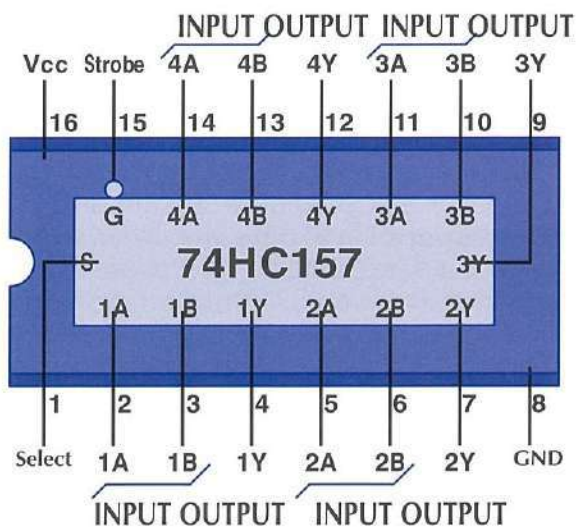
Questo circuito è un multiplexer digitale a otto ingressi di dati, un'uscita negata e tre terminali di controllo. Fa parte della famiglia TTL e dispone anche di un ingresso di abilitazione che deve essere a livello basso per poter selezionare uno degli ingressi.

## CI 74157

Questo circuito è realmente un multiplexer selettore che ha due ingressi di dati da quattro bit ognuno e un'uscita da quattro bit. Attraverso un bit di controllo, è possibile selezionare uno o l'altro gruppo di ingressi e determinare quali sono i quattro bit di ingresso che verranno portati all'uscita.



Circuito integrato 74HC157.

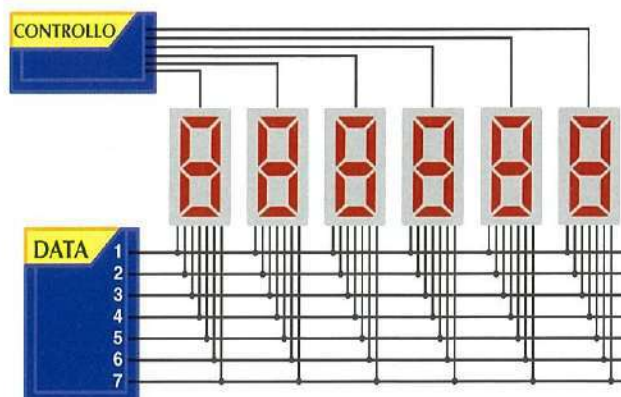


Multiplexer selettore 74HC157.

## Display multiplexato

Nei display, specialmente quelli a LED con molti digit, è possibile risparmiare su cablaggio e consumo illuminandoli alternativamente ad una frequenza così elevata che l'occhio umano non riesce a percepire quando si spengono, grazie alla persistenza dell'immagine sulla retina.

Ad esempio, per sei display da sette segmenti sono necessari 42 fili di connessione o piste di PCB, invece è possibile collegare sei display in parallelo e utilizzare solo sette fili e con altri sei selezionare il display che si illumina, allo stesso tempo inviare sulle sette linee l'ordine al display per fare illuminare i segmenti necessari a rappresentare il numero corretto.



Display multiplexato.





# Porte, livelli e ritardi

**N**ella teoria e nella pratica si lavora con i valori 1 e 0, tuttavia occorre tener presente anche altri parametri per fare in modo che i circuiti funzionino correttamente. Normalmente non si lavora con valori esatti di grandezze, pertanto bisogna conoscere con precisione fra quali margini di tolleranza si può utilizzare ogni grandezza. Non si tratta di un'operazione complicata, inoltre la padronanza di questi concetti è la base del successo di un progetto.

## Livelli logici

Quando si utilizza una determinata famiglia logica è necessario, innanzitutto, conoscere i dati generati da essa, compresi quelli che fanno riferimento al circuito integrato in questione, dato che potrebbero esserci alcune piccole differenze, magari poco importanti nel nostro caso, ma che potrebbero diventare necessarie nel caso si volessero sfruttare al massimo le prestazioni.

La prima cosa che dobbiamo conoscere è il livello al quale viene assegnato il valore 0 e a quale il valore uno. In generale e per la logica positiva l'uno si assegna al massimo della tensione di alimentazione, mentre lo zero a 0 V.

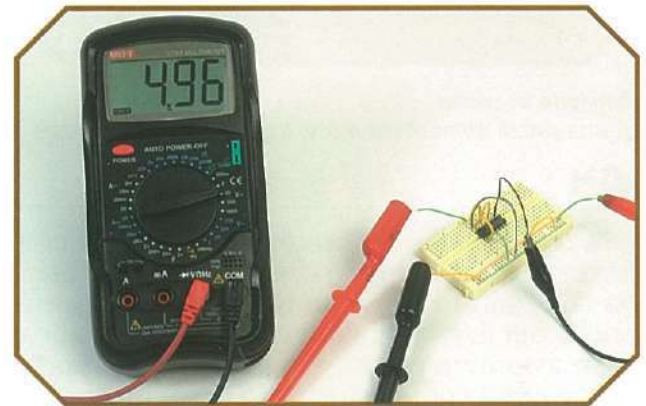
Dobbiamo anche definire il margine di variazione, e questo va fatto in modo generico per poter capire con facilità i dati che il costruttore fornisce nei cataloghi.

## VIL

È la tensione da applicare all'ingresso di una porta per fare in modo che questo livello sia considerato come uno zero logico. Specifica anche il livello massimo che questa tensione deve assumere per fare in modo che la porta, senza alcun problema, lo consideri uno zero. Si descrive inoltre come la tensione di ingresso massima per livello basso  $VIL_{max}$ .



Tensioni di ingresso e uscita di una porta logica.



Tensione di uscita di una porta con alimentazione da 5 V, livello alto.

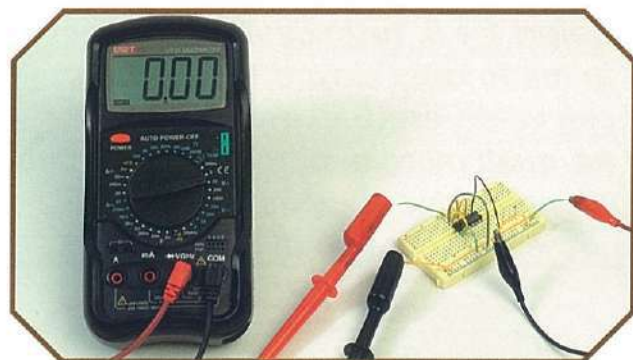
## VOL

È la tensione di uscita che deve fornire una porta per fare in modo che sia considerata come uno zero logico. Indica, inoltre, il livello massimo che questa tensione deve assumere perché sia considerata zero. La tensione di uscita massima per livello basso è chiamata  $VOL_{max}$ .

## Margini per livello basso

Se osserviamo l'immagine a fianco vediamo una differenza fra  $VIL$  e  $VOL$ . È permesso un margine maggiore sull'ingresso di una porta rispetto all'uscita di un'altra. Questo affinché si possa garantire che l'ingresso della porta venga considerato come zero – nonostante il livello del rumore normalmente presente – si possa sommare al livello dell'uscita e che quando il segnale arriva all'ingresso della porta, nonostante la presenza del rumore, sia considerato come uno zero.

La differenza  $VIL_{max} - VOL_{max}$  si chiama  $VNL$ , ovvero, livello di tensione di rumore permesso per livello basso (la lettera N deriva dal termine inglese "noise" = rumore).



Tensione di uscita  
di una porta alimentata a 5 V, livello basso.

## VIH

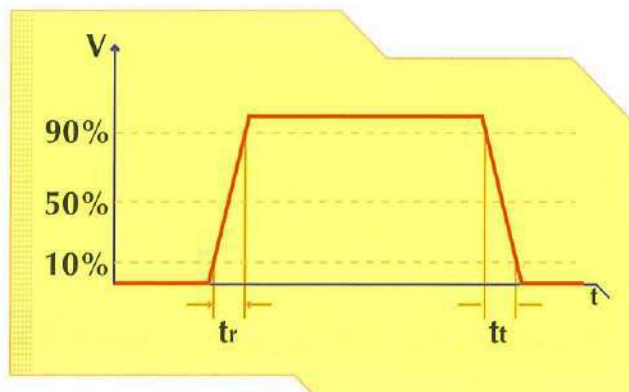
È la tensione da applicare all'ingresso di una porta per fare in modo che questo livello venga considerato come un uno logico. Si tratta anche del livello minimo che questa tensione deve assumere per fare in modo che la porta lo consideri come valore uno. Si definisce come tensione di ingresso minima per livello alto VIHmin.

## VOH

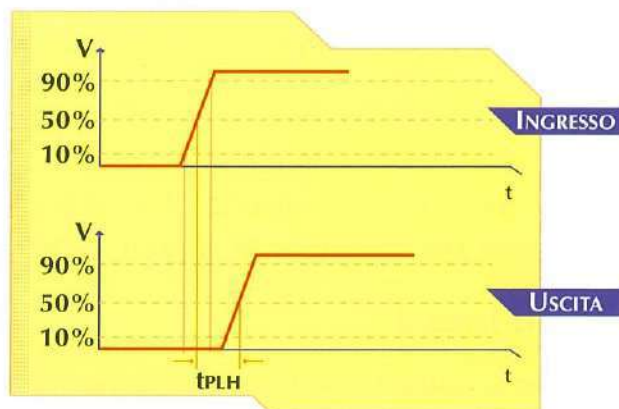
È la tensione di uscita che deve fornire una porta per generare un uno logico. È anche il livello minimo che questa tensione deve avere perché sia considerato di valore uno. La tensione di uscita minima per livello alto è chiamata VOHmin.

## Margini per livello alto

Così come per il livello basso, anche per il livello alto vediamo che non coincidono i livelli di ingresso e di uscita di una porta. Se torniamo



Ritardo in un impulso reale.

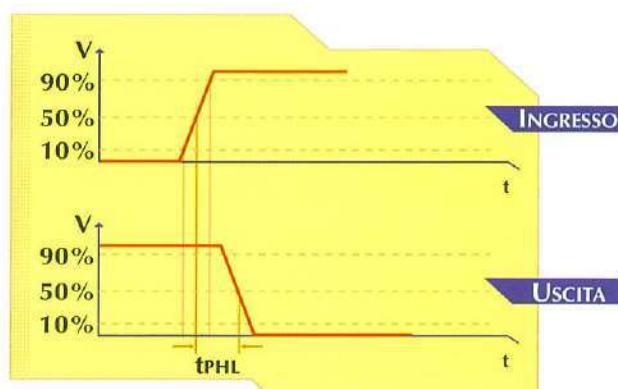


Ritardo di propagazione ON.

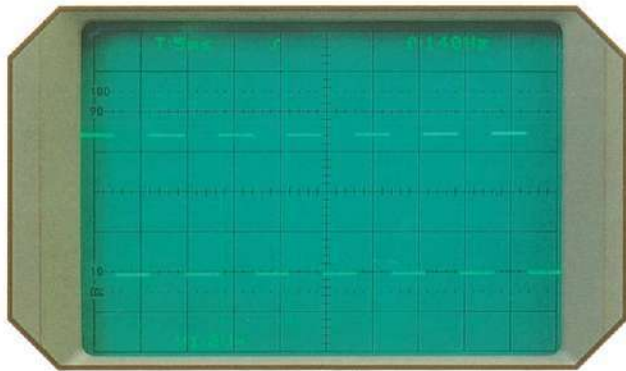
a osservare l'immagine della pagina precedente vediamo una differenza fra VOH e VIH. Il margine di ingresso della porta è maggiore di quello permesso all'uscita. In questo modo è possibile accettare un certo livello di rumore aggiunto all'uscita di una porta, senza che questo causi problemi, ovvero, facendo in modo che il segnale si mantenga all'interno del margine ammesso all'ingresso di un'altra porta e, quindi, venga considerato come un uno logico. La differenza VIHmin - VOHmin è chiamata VNH, cioè, il livello di rumore di tensione permesso a livello alto.

## Rumore

Se osserviamo le caratteristiche riportate nei Data Sheet dei costruttori, possiamo constatare che le diverse famiglie di circuiti integrati hanno comportamenti differenti rispetto al rumore, basta comparare i dati di VNH e di VNL. La famiglia CMOS della serie 4000 che utilizzeremo, presenta una buona immunità al rumore, come verificheremo nella pratica al momento opportuno.



Ritardo di propagazione OFF.



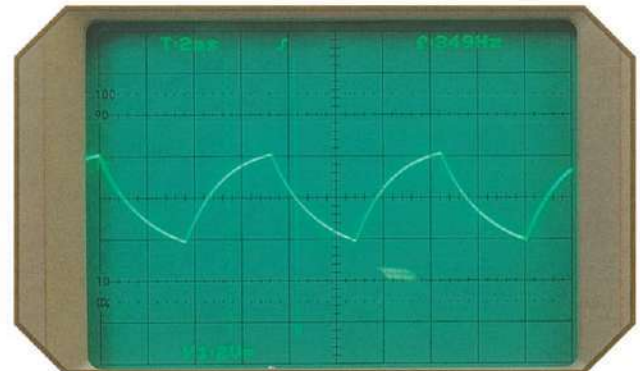
Treni di impulsi.

## L'impulso ideale

Nell'elettronica digitale si lavora con impulsi, ovvero, il segnale cambia livello passando al livello logico uno o zero a seconda della necessità, per fare in modo che il bit abbia valore 1 oppure 0. Normalmente gli impulsi vengono rappresentati con dei parallelogrammi con fronti dritti. Se considerassimo reale il tempo impiegato da un circuito nel passare da zero a uno o da uno a zero, il valore sarebbe zero, cioè, istantaneo e la velocità dei circuiti sarebbe di conseguenza, infinita. Tuttavia, per quanto i circuiti siano realmente molto rapidi, la loro velocità non è infinita, ma vengono generati dei piccoli "ritardi" che identificano precisamente la velocità con cui possono lavorare i circuiti digitali.

## Ritardi

Utilizzeremo ora un impulso per definire i tempi che impiega un segnale per passare dallo stato logico zero allo stato logico uno, e vi-



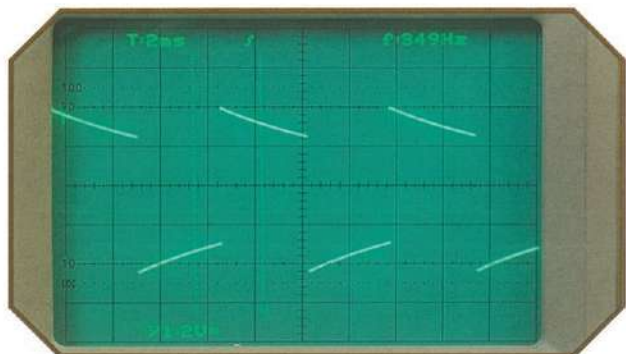
Impulsi deformati da una rete RC.

ceversa, per passare dallo stato logico uno allo stato logico zero. La rappresentazione più utilizzata e più vicina alla realtà è quella trapezoidale. Sull'asse verticale si rappresenta la tensione e su quello orizzontale il tempo. In questa rappresentazione il livello massimo è il massimo di tensione che si assegna al valore uno logico, e lo zero di tensione si assegna allo zero logico. Sull'asse verticale verranno segnate chiaramente le tensioni massima e minima, oltre ai valori 10%, 50% e 90% della stessa.

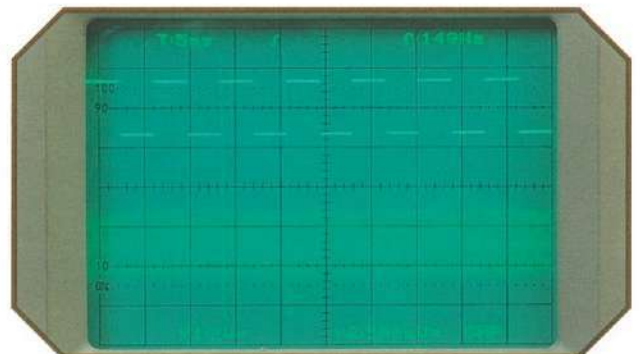
Si disegna un segnale che rappresenta un impulso, ovvero, un segnale che passa dal livello basso al livello alto, si mantiene al livello alto per un certo periodo per poi tornare al livello basso. Se si osserva la figura, i cambi di livello hanno un certo ritardo in entrambi i casi.

## Tempo di salita

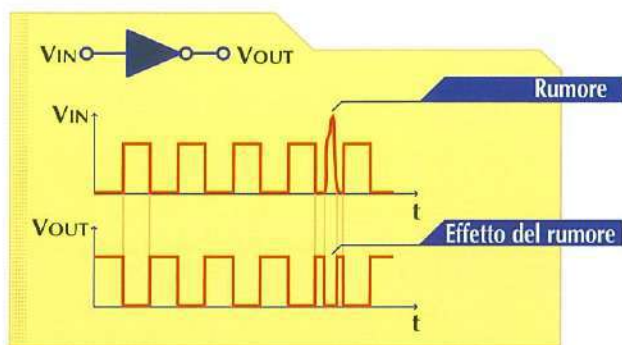
Si tratta del periodo di tempo in cui il segnale passa dal livello basso al livello alto, viene deno-



Impulsi deformati da una rete CR.



Treni di impulsi da 5 V e rumore captato da un cavo da 2 m.



*Il rumore può creare segnali imprevisti.*

minato tempo di salita il tempo che impiega la tensione dal momento in cui raggiunge il 10% fino a quando raggiunge il 90%. Si rappresenta come  $t_r$  (dall'inglese rise time).

### Tempo di discesa

Definisce il tempo che impiega il segnale per passare dal livello alto al livello basso, ed è generalmente diverso da quello di salita. Si tratta del tempo impiegato dal segnale per passare dal 90% del valore massimo di tensione al 10%. Viene chiamato  $t_f$  (dall'inglese "fall time").

### Ritardo di propagazione

Il ritardo di propagazione è il tempo impiegato dall'uscita per cambiare stato, in seguito al cambio di stato dell'ingresso. Consideriamo il tempo di propagazione per il caso in cui l'uscita passa dal livello basso al livello alto.

Questo ritardo si chiama  $t_{PLH}$ . I costrutto-

ri misurano questo tempo quando la tensione passa attraverso il 50%. Analogamente, il ritardo di propagazione è quello necessario per fare in modo che l'uscita passi dal livello alto al livello basso, in questo caso si chiama  $t_{PHL}$  (cioè passaggio da High a Low).

Per poter aumentare la velocità dei propri circuiti, i costruttori sono costantemente impegnati a ridurre proprio questi tempi di ritardo.

### Rete RC

Un modo molto semplice di osservare come una capacità può modificare un impulso è quello di applicare un impulso teoricamente perfetto ad una rete RC o CR e osservare sull'oscilloscopio come lo stesso si deforma e "ritarda".

### Protezione contro il rumore

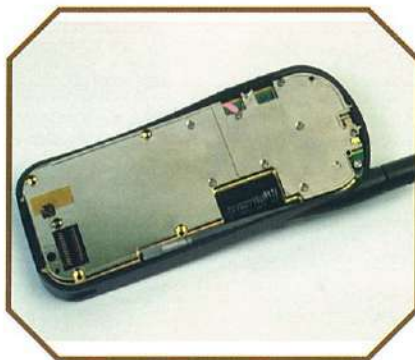
Il rumore fa parte dei circuiti ed è presente in tutte le parti, quindi, è necessario cercare di attenuarlo in modo da ridurlo a livelli tali da non arrecare alcun danno. Il rumore raggruppa un grande numero di disturbi, quindi si può affermare che è il motivo di maggior preoccupazione in un sistema digitale.

Si considera rumore qualsiasi disturbo non provocato intenzionalmente, che faccia cambiare in modo indesiderato lo stato dell'uscita di un circuito.

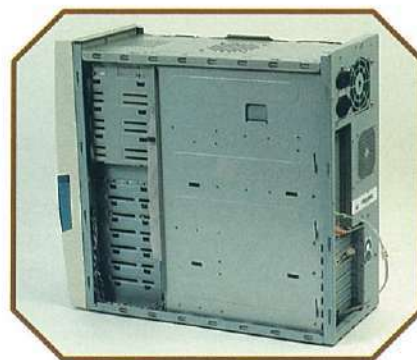
Esistono molte tecniche di protezione contro il rumore, la più semplice è la schermatura di cavi e circuiti.



*Cavo schermato.*



*Schermatura di un circuito.*



*Schermatura di un PC.*



## Conversione A/D D/A

**L**e grandezze che si utilizzano nel mondo reale sono, generalmente, di natura analogica. Tuttavia oggi si lavora come se tutto fosse digitale.

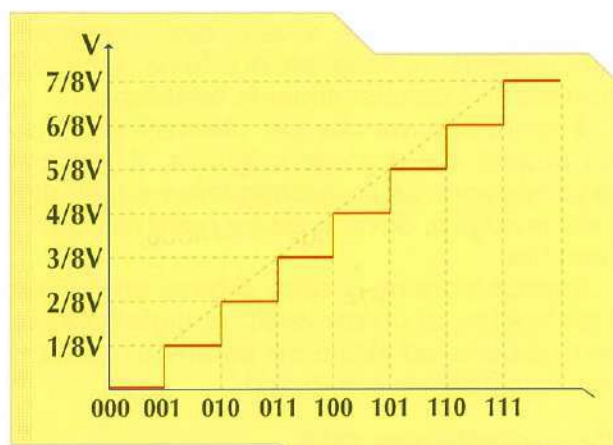
Per questo motivo, prima di iniziare gli esperimenti, bisogna aver ben chiaro che le grandezze analogiche continueranno a essere tali, però è possibile trattarle in modo digitale, "digitalizzandole". Si tratta di un processo delicato che deve essere realizzato in modo corretto per poter recuperare successivamente la grandezza analogica. Dopo che la grandezza – per capirci meglio – il segnale è stato convertito in digitale, presenta il vantaggio di poter essere trattato con potenti strumenti, alcuni dei quali verranno utilizzati anche da noi. Infine, dovremo riportare il segnale nel suo stato analogico, ma con caratteristiche analogiche, ovvero un segnale di qualità perfettamente utilizzabile.

### Convertitori

I circuiti o strumenti che convertono un segnale o una grandezza analogica in digitale si chiamano convertitori analogico/digitale e vengono indicati come A/D o anche come ADC. I circuiti o gli strumenti che realizzano la funzione inversa si chiamano convertitori digitale/analogico, o DAC.

### Audio digitale

Facciamo un esempio molto attuale che aiuterà a comprendere bene i concetti: l'audio digitale. Attualmente possiamo considerare una canzone come un file di dati scritto in una memoria, su un CD o DVD, con un determinato formato. Per poter riprodurre un file musicale memorizzato è sufficiente uno strumento, la cui parte più importante è il convertitore, che partendo dal segnale digi-

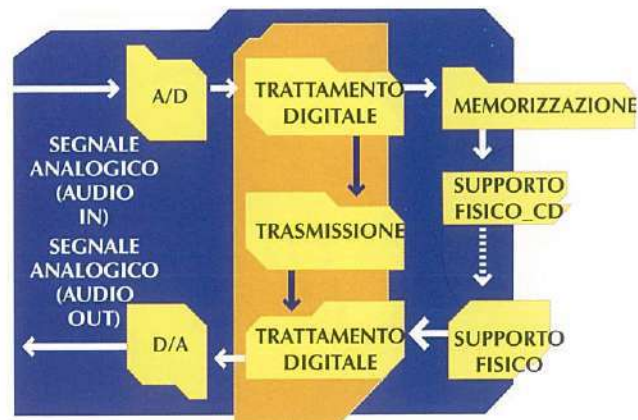


A ogni codice d'ingresso corrisponde un livello di tensione di uscita.

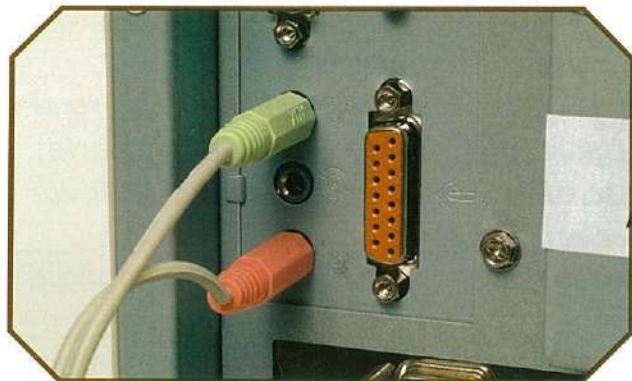
tale fornito dagli altri circuiti dello strumento, ottiene un segnale analogico. Questa parte si chiama convertitore digitale/analogico DAC. Tutti siamo testimoni della qualità del suono che forniscono questi strumenti, frutto dello sforzo e del lavoro di molte persone.

### La qualità

Quasi senza rendercene conto, abbiamo fatto riferimento alla qualità. Un modo per verificare la qualità del segnale digitale è confrontarlo con un segnale uguale ma analogico. La prima cosa da fare è scegliere un segnale analogico noto e di alta qualità, convertirlo in digitale, memorizzarlo, ad esempio su un CD, e poi riprodurlo. Questo segnale, originariamente analogico, dopo aver



Sequenza riassuntiva del trattamento digitale dei segnali analogici.



*I collegamenti di ingresso e di uscita audio di un PC utilizzano segnali analogici.*

subito un processo digitale, deve avere la stessa qualità dell'originale, ovvero, deve rappresentare fedelmente il segnale di origine, a meno di non averlo intenzionalmente modificato.

È facile dedurre che per ottenere una buona qualità del segnale originale, il processo di conversione analogico/digitale e quello digitale/analogico, devono essere realizzati correttamente.

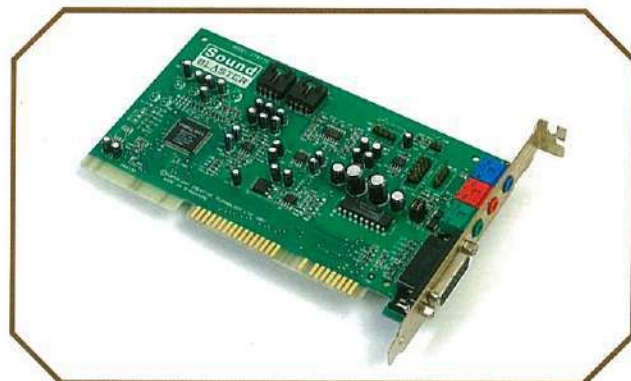
Vedremo ora un circuito di base, però molto significativo, di un convertitore digitale/analogico, assieme ad alcuni dei parametri utilizzati per classificare questi circuiti.

## Convertitore D/A

Il convertitore D/A fornisce, alla sua uscita, un segnale analogico in base alle differenti word logiche che riceve sul suo ingresso a una determinata velocità. Generalmente utilizza 8 o 16 bit in parallelo a una velocità, per esempio, di



*L'audio è puramente analogico però si può trattare con circuiti e strumenti digitali.*



*Da diverso tempo tutti i PC sono dotati di circuiti audio.*

44.100 Hz. A ogni word o byte applicato all'ingresso, si ottiene un corrispondente livello analogico all'uscita.

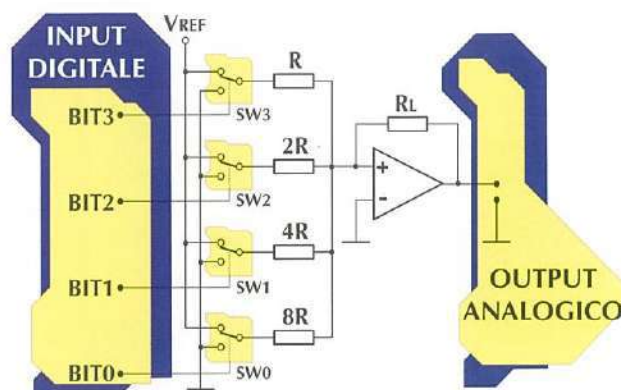
L'uscita analogica può essere sotto forma di corrente o di tensione. Il passaggio da corrente a tensione si realizza con un amplificatore operazionale.

Dobbiamo definire, fra gli altri parametri, il fondoscala, cioè, la massima tensione analogica possibile all'uscita, che si ottiene con il codice digitale più alto applicato all'ingresso.

L'uscita dei convertitori digitale/analogico può essere unipolare o bipolare. Se l'uscita è unipolare, il livello del segnale analogico di uscita va da 0 a un determinato valore, solitamente positivo, che è il fondoscala. Se l'uscita è bipolare, il livello del segnale analogico di uscita può essere positivo, in questo caso il fondoscala si definisce con un limite positivo e un altro negativo.



*Gli attuali strumenti audio utilizzano la memorizzazione digitale.*



Schema di un convertitore DIA di base.

## Parametri

Oltre a quelli già menzionati, ci sono altri parametri molto interessanti, anche se alcuni di essi, al momento, possono risultare un po' complicati da capire. È necessario conoscerli per poter scegliere, in ogni situazione, il più adeguato.

### Risoluzione:

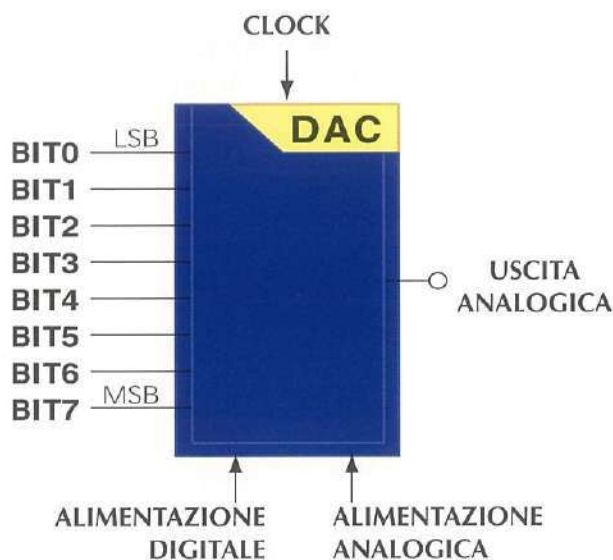
È la più piccola variazione che si produce sulla tensione di uscita quando il codice d'ingresso incrementa o diminuisce di un BIT.

### Tempo di assestamento:

È il tempo che passa da quando la word digitale si applica all'ingresso fino a quando l'uscita arriva al livello analogico corrispondente.

### Uscita in modo corrente:

Si dice che l'uscita è di questo tipo quando la



Collegamenti fondamentali di un DAC.

corrente analogica di uscita è proporzionale alla word digitale di ingresso.

### Uscita in modo tensione:

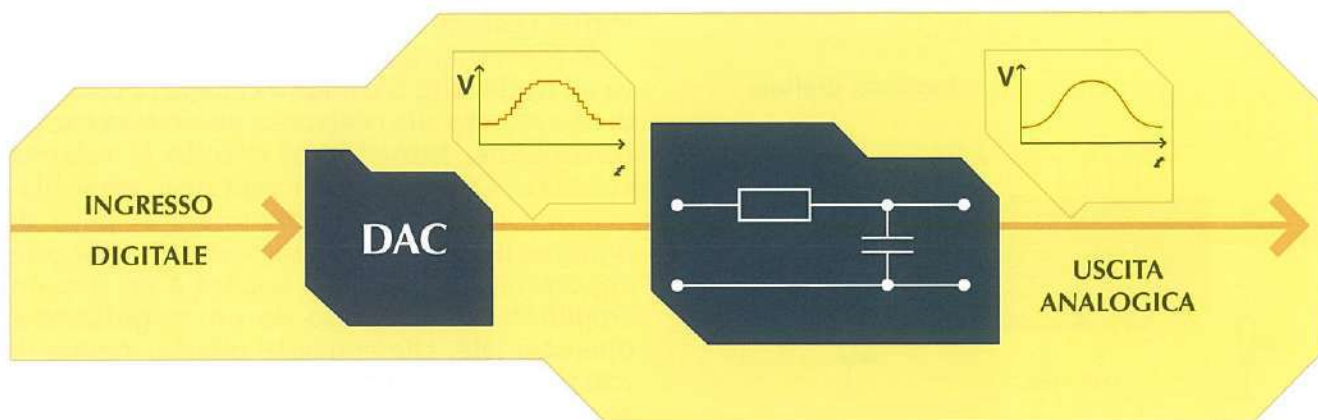
L'uscita analogica è una tensione proporzionale al codice digitale applicato all'ingresso. Questo tipo di convertitore è più lento di quello in corrente.

### Margine dinamico:

Il margine dinamico indica il massimo margine di tensione o di corrente dell'uscita.

### Codici di ingresso:

Sono i codici digitali di ingresso che può gestire il convertitore.



Un semplice filtro passabasso può eliminare praticamente il ripple della conversione.



### Errore di spostamento:

È la tensione di uscita quando la word digitale di ingresso corrisponde al valore analogico zero.

### Errore di fondoscala:

È la deviazione della pendenza della funzione di trasferimento dal suo valore ideale. Indica la deviazione per il valore massimo di tensione di uscita.

### Errore di linearità:

È la deviazione dalla caratteristica di trasferimento ideale.

### Monotonicità:

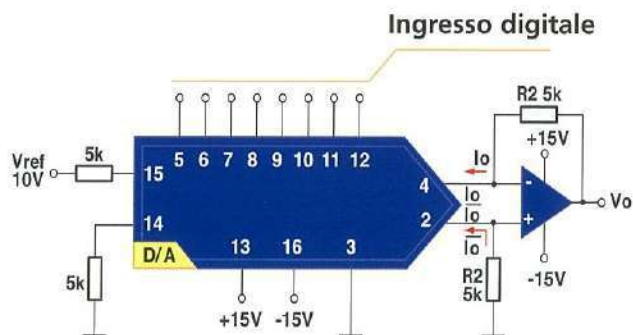
Significa che, a fronte di un incremento della word digitale, l'uscita analogica deve aumentare.

## Convertitore D/A di base

Esistono molti circuiti di conversione D/A, tuttavia, quello chiamato di base è molto significativo, poiché indica in modo abbastanza intuitivo come si può realizzare un convertitore D/A.

Ingresso digitale	Uscita analogica
0 0 0	0,00
0 0 1	1,25
0 1 0	2,50
0 1 1	3,75
1 0 0	5,00
1 0 1	6,25
1 1 0	7,50
1 1 1	8,75

Tensione di uscita di un convertitore DAC da 3 bit con fondoscala da 10 volt.



Schema tipico di un DAC.



Le stesse grandezze si possono rappresentare in diversi modi, attraverso strumenti analogici o digitali.

Se osserviamo lo schema elettrico della pagina precedente, possiamo vedere i seguenti elementi:

- Un insieme di commutatori, il cui numero è uguale al numero di bit, in questo caso 4.
- Una tensione VREF di riferimento che deve essere più precisa e stabile possibile.
- Una rete di resistenze ponderata, per scomporre la word digitale in diversi livelli di tensione o di corrente.
- Un circuito amplificatore operazionale di uscita che somma le correnti ottenute e fornisce alla sua uscita una tensione equivalente e proporzionale al dato digitale di 4 bit applicato all'ingresso.

Quando il dato digitale si applica all'ingresso di controllo di ciascun commutatore, questi ultimi, a seconda se il bit applicato è un uno o uno zero, faranno circolare o meno la corrente.

Se a un commutatore arriva uno zero, cioè un livello basso, non circolerà corrente, poiché il commutatore verrà collegato a massa e non immetterà corrente nel circuito.

Al contrario, quando un commutatore riceve un livello alto si chiude e collega la tensione di riferimento alla resistenza ponderata che gli corrisponde, fornendo al circuito la relativa corrente. Con questi commutatori applichiamo una corrente proporzionale al peso di ognuno, in modo che la somma della corrente apportata da ogni bit si applica a un circuito amplificatore costituito da un amplificatore operazionale, che converte questa somma di corrente in una tensione corrispondente alla word digitale applicata all'ingresso del convertitore digitale/analogico.





## Conversione A/D

**A**bbiamo già parlato di alcune caratteristiche dei convertitori digitale Analogico DA. Ora vedremo alcuni aspetti dei convertitori AD, ovvero quelli che realizzano la funzione inversa convertendo un livello di tensione in un codice digitale.

### Campionamento e ritenzione

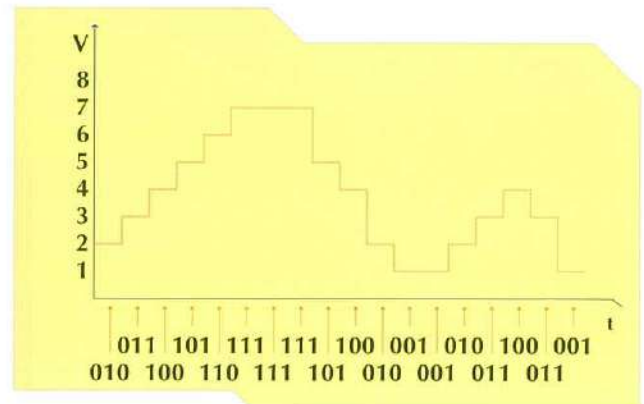
Come abbiamo appena detto, si tratta di convertire un livello di tensione, che è una grandezza analogica, in un codice digitale. Questa procedura è utile solo per alcuni tipi di applicazioni come, ad esempio, per misurare la tensione, la temperatura o qualsiasi altra grandezza analogica che si possa convertire in una tensione.

Per altri tipi di applicazioni, quando si vuole digitalizzare il segnale elettrico, fornito da un preamplificatore che contiene l'informazione audio ricevuta da un microfono, è possibile utilizzare la conversione, ma è necessario prendere periodicamente, nella maggiore quantità possibile, campioni di segnale. Una delle frequenze di campionamento utilizzate nel settore audio è nell'ordine dei 44 KHz, in pratica si esegue un campionamento ogni due microsecondi. Dobbiamo tener presente che il convertitore impiega un certo tempo nel realizzare la conversione, quindi sono necessari convertitori veloci per poter utilizzare frequenze di campionamento alte.

Per fare in modo che il convertitore funzioni e possa fornire correttamente il codice corrispondente a ogni campione, il livello di tensione deve essere mantenuto all'ingresso del convertitore fino alla fine del processo di conversione, dopodiché si può dare l'ordine di iniziare la



Il primo passo è prendere campioni del segnale.



La codificazione assegna i codici disponibili.

conversione del successivo campione in codice digitale.

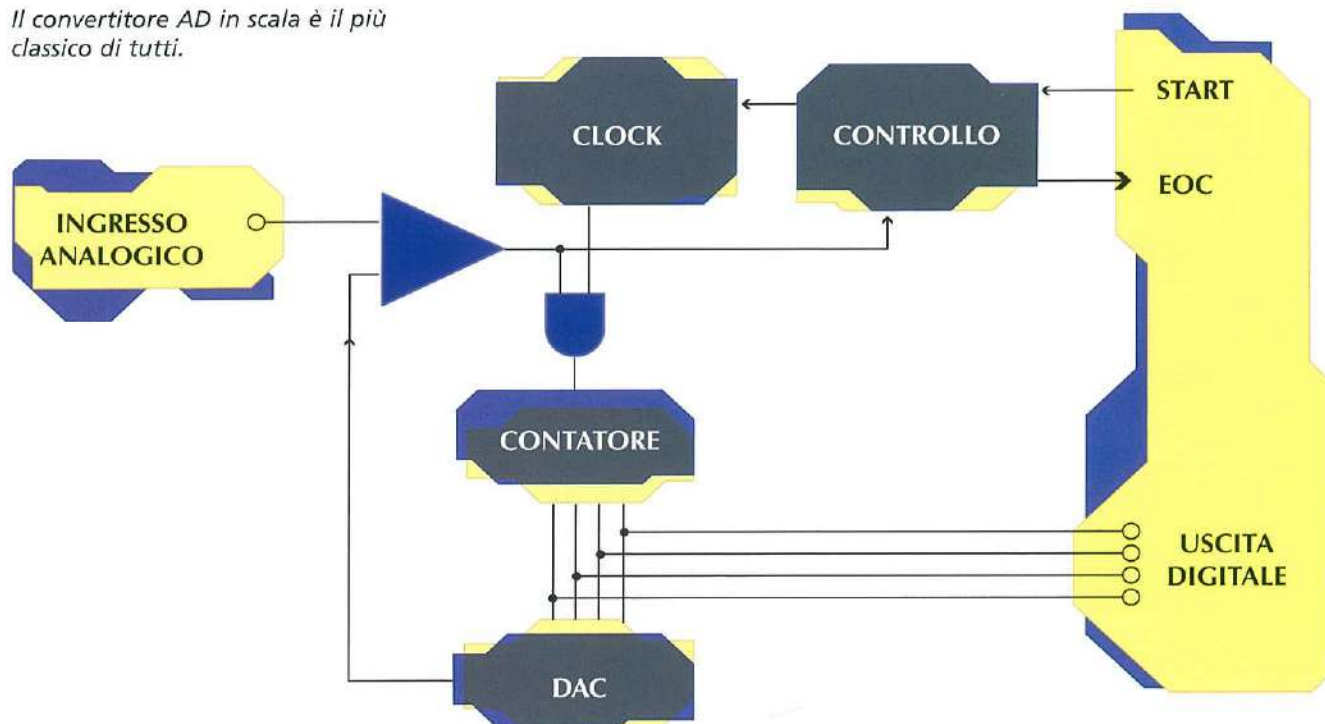
All'ingresso del convertitore bisogna realizzare un circuito di campionamento e ritenzione, che in inglese si chiama Sample and Hold.

### Risoluzione

La velocità di campionamento per i dispositivi audio deve essere molto alta, come minimo il doppio della massima frequenza audio del segnale che si sta campionando. È necessario avere una buona risoluzione, cioè, molti bit di uscita. È facile capire che con soli 2 bit possiamo rappresentare 4 valori di tensione, con 3 bit 8 valori, con 4 bit 16 valori, e così via. Se la risoluzione con cui si esegue un campionamento è cattiva, il segnale riprodotto può essere molto differente da quello originale. In altre parole, se supponiamo che il massimo segnale applicato all'ingresso di un convertitore è di 10 volt e utilizziamo un convertitore che ha solo 4 bit, avremo 16 livelli di tensione; facendo un calcolo approssimato, dividiamo 10 per 16 e il risultato è di 0,625 volt, cioè tutti i livelli di tensione di ingresso tra 6,25 e 6,875 volt saranno rappresentati, al momento della riproduzione, da un medesimo codice. Se invece, eleviamo a 8 il numero dei bit, divideremo 10 volt per 256, riducendo



Il convertitore AD in scala è il più classico di tutti.



il salto di tensione a 0,04 volt e disponendo, in questo caso, di 16 valori di tensione da inserire come 16 valori intermedi tra 6,25 e 6,875, quindi il segnale riprodotto sarà molto più simile a quello originale.

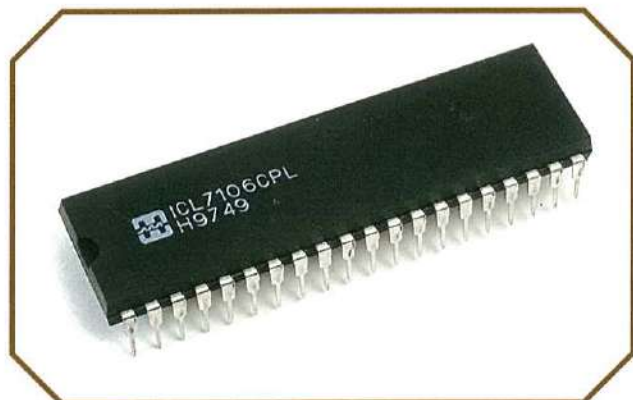
Dobbiamo tener conto che aumentando il numero di bit aumenta anche il tempo di conversione. Esiste sul mercato una grande varietà di convertitori, sia in tecnologia classica che in versione sigma delta.

La varietà è così grande che quando nasce una necessità particolare bisogna ricorrere ai

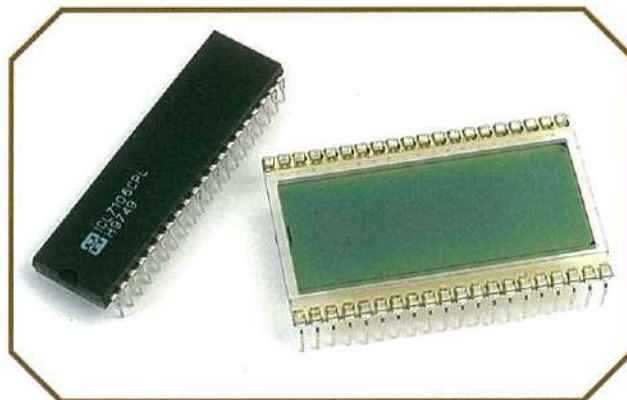
costruttori; è molto interessante visitare le pagine Internet di Dallas semiconductor/MAXIM che presenta una grande varietà di modelli, accompagnata dai Data Sheet delle caratteristiche, e soprattutto dalle note esplicative.

## Parametri

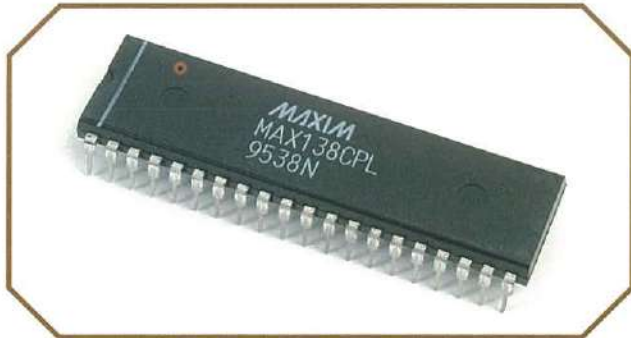
Vedremo ora come si elencano alcuni dei parametri che si utilizzano nei convertitori analogico digitali.



L'integrato ICL 7106 è un circuito molto noto e contiene un convertitore AD, utilizzato per costruire multimetri digitali.



L'integrato ICL 7106 contiene anche i driver necessari per pilotare direttamente un display LCD da 3 e 1/2 digit.



Il MAX 138CPL è una versione migliorata di ICL 7106.

**Risoluzione**

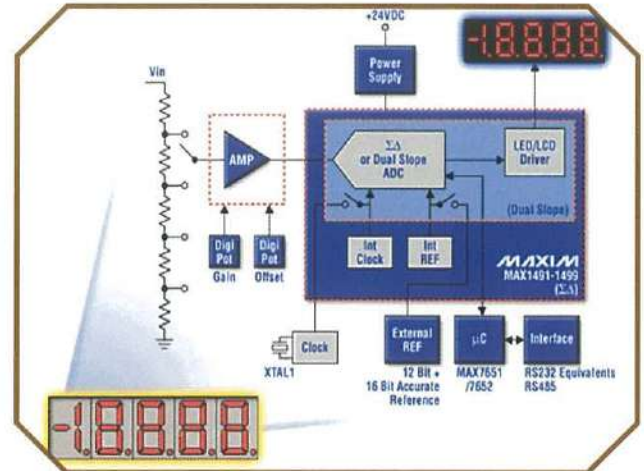
È la minima variazione di tensione analogica d'ingresso che fa variare la word digitale di uscita al valore immediatamente successivo.

**Tempo di conversione**

È il tempo che trascorre da quando inizia la conversione della tensione analogica a quando si ottiene il segnale digitale.

**Codice di ingresso**

Sono i codici binari con cui lavora il convertitore analogico digitale.



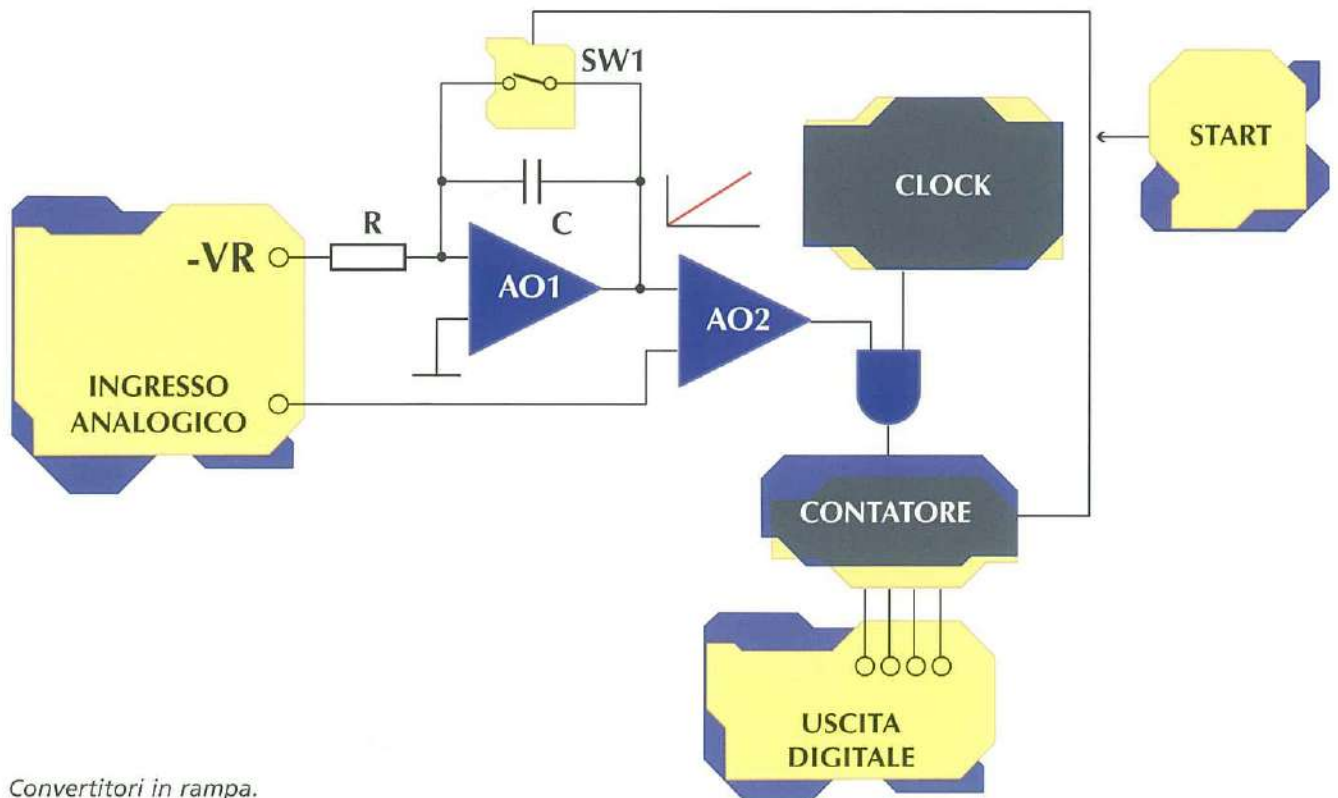
Con i nuovi convertitori di Dallas-Maxim si facilita la costruzione di strumenti e il loro collegamento con sistemi esterni.

**Errore di spostamento**

È il valore di tensione analogica di ingresso per ottenere la word con codice 000...0.

**Errore di fondoscala**

Lo produce l'amplificatore differenziale di uscita che può variare la retta di trasferimento.



Convertitori in rampa.



### Segnali di controllo

Sono segnali tipici che vengono definiti per i convertitori AD:

- EOD termine della conversione: indica che il convertitore ha ottenuto il codice digitale corrispondente al livello del segnale analogico.
- Inizio della conversione: è l'ordine con cui i circuiti restano configurati per poter iniziare la conversione.
- Clock.

### Conversione A/D in scala

Questo convertitore corrisponde a uno schema classico ed è riportato in tutte le pubblicazioni; è piuttosto lento e ha un funzionamento facile da capire. Il livello del segnale analogico che fornisce il circuito di campionamento e ritenzione, o il segnale in continua, si applica all'ingresso di un amplificatore differenziale configurato come comparatore. Quando arriva il segnale di inizio della conversione il contatore viene impostato a zero, si attiva il clock e inizia ad avanzare il conteggio del contatore, però, come possiamo vedere osservando il diagramma a blocchi della pagina precedente, l'uscita del contatore entra in un convertitore digitale analogico, il quale fornisce un livello di tensione che sale man mano che avanza il conteggio del contatore. Questa tensione si compara con la tensione analogica d'ingresso e quando la raggiunge cambia lo stato di uscita del comparatore e viene inviato il segnale al circuito di controllo per fermare il contatore. L'uscita del contatore è il codice digitale assegnato come corrispondente al livello che ha

causato l'attivazione del comparatore, ed è uguale al livello di tensione analogico applicato all'ingresso. In questo tipo di convertitore, il tempo di conversione dipende dal livello del segnale analogico applicato all'ingresso.

### Convertitori in rampa

Anche questo modello di convertitore è molto classico, utilizza un comparatore (AO2) e il segnale analogico è applicato all'ingresso non invertente dello stesso. Il circuito di controllo, oltre a impostare a zero il contatore, fa in modo che inizi la rampa di tensione dell'integratore, controllando, a questo scopo, l'interruttore SW. La rampa di tensione così ottenuta si applica all'altro ingresso del comparatore e, quando si raggiunge la tensione analogica d'ingresso, l'uscita del comparatore cambia livello e si ferma il contatore. Il dato presente in quel momento sul contatore è il codice digitale che viene assegnato alla tensione di ingresso.

### I convertitori Sigma-Delta

Questi convertitori hanno un principio di funzionamento piuttosto complicato da spiegare, quindi non entreremo nel dettaglio, però è necessario sapere che esistono. Con questi convertitori, molto facili da integrare, è possibile ottenere un'alta risoluzione e un basso costo. Inoltre, questi circuiti integrati dispongono di funzioni aggiuntive per facilitare il loro collegamento al resto della circuiteria del dispositivo in cui sono utilizzati.

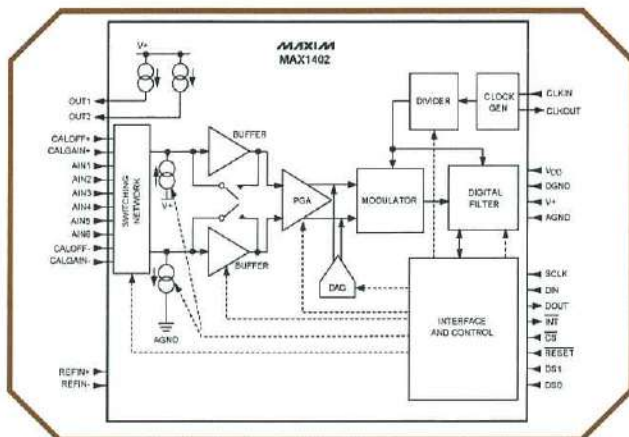


Diagramma a blocchi del convertitore MAX 1402 in tecnologia Sigma-Delta.

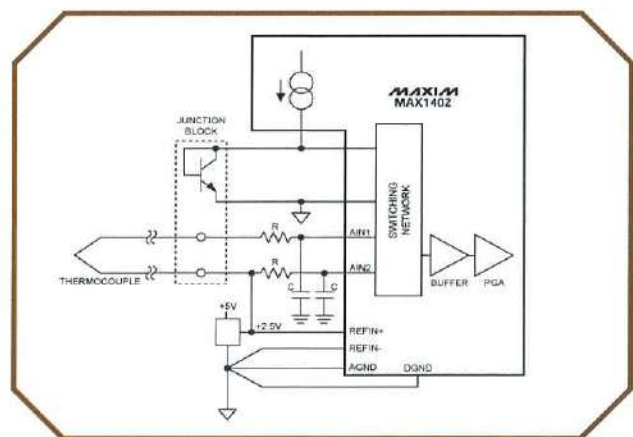


Diagramma semplificato di un convertitore D/A di Dallas-Maxim per indicatore di temperatura.



## Latch

**Q**uesto nome si applica a un bistabile, ovvero un circuito la cui uscita può avere due livelli. Si tratta di un circuito sequenziale che utilizza un circuito combinatoriale e uno o più elementi di memoria.

Un circuito combinatoriale è realizzato grazie a una combinazione di porte logiche, e la sua uscita o le sue uscite forniscono un determinato risultato sotto forma di uno o zero, che deriva unicamente dalla combinazione applicata all'ingresso, indipendentemente dallo stato precedente.

Il circuito sequenziale, oltre alla combinazione applicata all'ingresso, dipende anche dall'informazione precedentemente memorizzata sul circuito, ovvero, può variare con il tempo ed essere determinata dallo stato precedente del circuito.

### Bistabili

I bistabili si dividono in due categorie principali, quelli che si attivano per livello e quelli che si attivano per fronte. Il fronte è la parte quasi verticale dell'impulso e può essere di salita o di discesa.

Benché sia facile confondersi, ricordiamo che la denominazione flip-flop si applica a bistabili attivati tramite un fronte e latch a quelli attivati tramite livello. Per prima cosa vedremo il latch.

Come abbiamo detto, un bistabile è un circuito sequenziale ed è parte integrante di registri, contatori, memorie, ecc.; in realtà è una cella elementare di memoria.

### Sincrono, asincrono

Anche questa classificazione è piuttosto facile da comprendere. In un bistabile asincro-

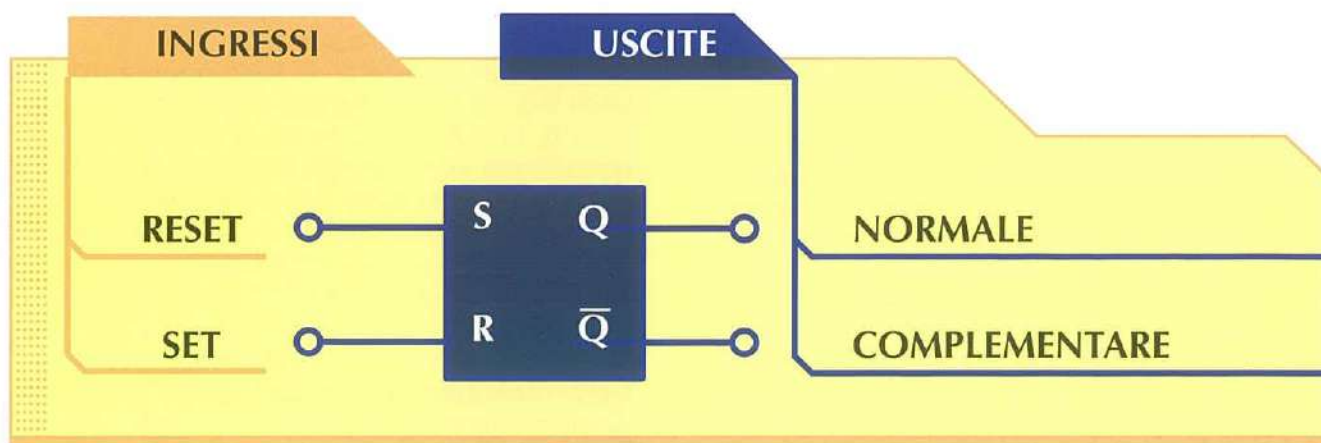
no l'uscita si modifica nel momento in cui si produce una variazione all'ingresso, e questo può accadere in qualsiasi momento.

Nei bistabili sincroni l'uscita e lo stato interno sono campionati in determinati istanti che dipendono dal segnale applicato all'ingresso del clock. È ovvio quindi che hanno bisogno di un ingresso di clock.

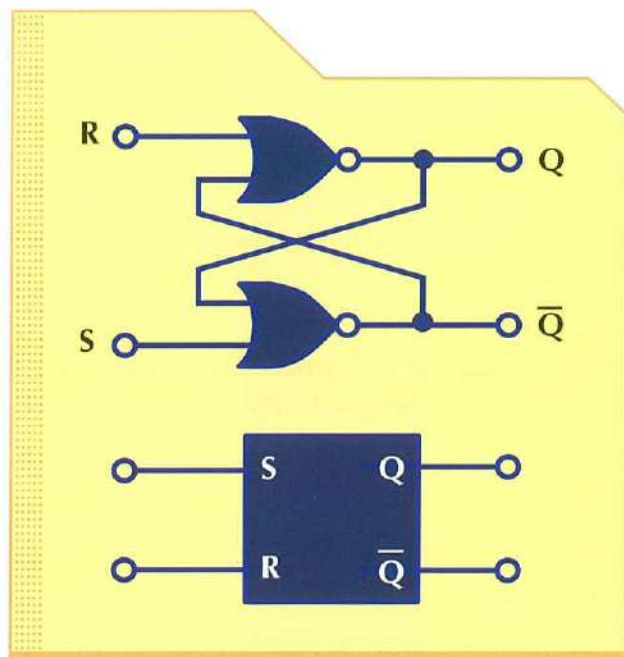
### Bistabile base

Il circuito logico ha due stati stabili denominati SET o attivazione, e RESET o disattivazione, i quali possono essere mantenuti in modo indefinito, quindi permettono di memorizzare un bit.

I bistabili si realizzano con porte logiche e osservando lo schema, ciò che più li differenzia da un circuito combinatoriale è che l'uscita in qualche modo è collegata all'ingres-



Terminali principali di un bistabile.



Bistabile R-S con porte NOR.

S	R	$Q_n$	$/Q_n$	
0	0	$Q_{n-1}$	$/Q_{n-1}$	Non cambia
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	Proibito

Bistabile R-S con porte NOR.

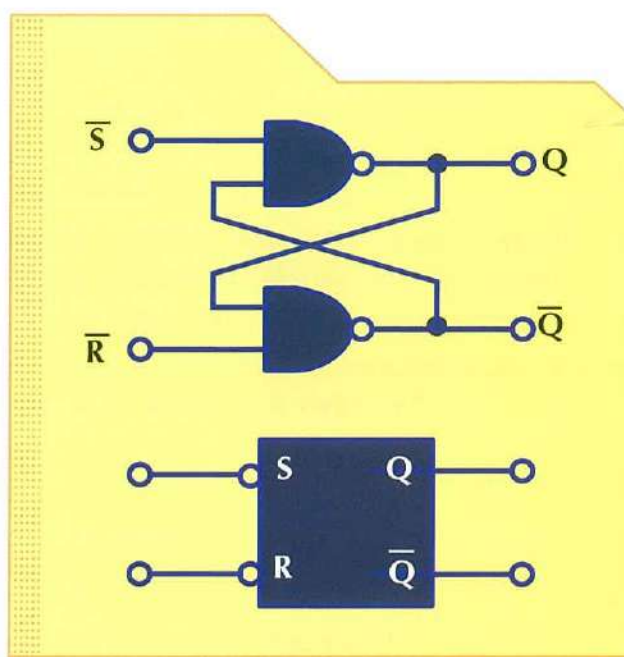
so precedente. Normalmente dispongono di un'uscita e della stessa uscita negata.

### Bistabili sensibili al livello o latch

Esistono due tipi di latch R-S che dipendono dal tipo di porte con cui sono stati implementati, anche se generalmente vengono entrambi indicati con la denominazione di bistabile.

#### Bistabile R-S con porte NOR

Il bistabile, o latch, R-S (Reset-Set), ha l'ingresso attivo a livello alto, un'uscita identificata con la lettera Q e la sua uscita complementare  $/Q$ . Nello schema possiamo osservare che è



Bistabile R-S con porte NAND.

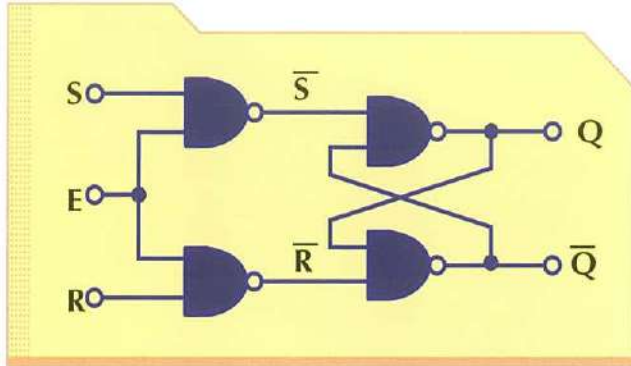
$/S$	$/R$	$Q_n$	$/Q_n$	
0	0	1	1	Proibito
0	1	1	0	SET
1	0	0	1	RESET
1	1	$Q_{n-1}$	$/Q_{n-1}$	Non cambia

Bistabile R-S con porte NAND.

composto da due porte NOR. La cosa più interessante è che l'uscita di una porta si collega all'ingresso dell'altra porta.

Descriviamo quindi il funzionamento del latch R-S. Se i due ingressi del latch, R e S, sono a 0, l'uscita del circuito si manterrà nello stato in cui era; quando l'ingresso R del latch passa a 1 e l'ingresso S rimane a 0, l'uscita Q passa a 0 indipendentemente dal suo stato precedente e a sua volta l'uscita negata  $/Q$  passerà a 1, il latch quindi passerà allo stato di Reset.

A questo punto, se R torna a 0 il circuito entrerà nuovamente nel suo modo memoria. Analogamente, se S passa a 1 e R rimane a 0, l'uscita negata  $/Q$  passa a 0 quindi l'uscita Q passa a 1. Riassumendo:



Bistabile R-S con porte ingresso di abilitazione.

E	S	R	Q <sub>n</sub>	/Q <sub>n</sub>	
1	0	0	Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia
1	0	1	0	1	RESET
1	1	0	1	0	SET
0	X	X	Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia

Bistabile R-S con porte ingresso di abilitazione.

- Quando si attiva (passa a 1) l'ingresso RESET, Q passa a 0.
- Quando si attiva (passa a 1) l'ingresso SET, Q passa a 1.
- Quando gli ingressi non cambiano (entrambi a 0) l'uscita non cambia.
- Quando si attivano i due ingressi (S a 1 e R a 1) il circuito non funziona correttamente, si tratta di una combinazione da evitare. Il latch R-S con porte NOR, ovvero, con gli ingressi at-

tivi a 1, è contraddistinto dal simbolo che vediamo nella figura della pagina precedente.

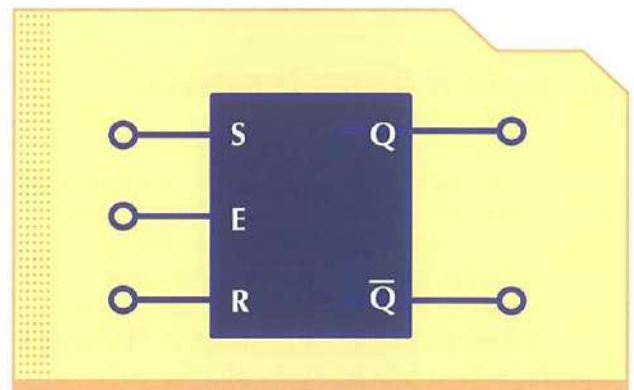
### Bistabile R-S con porte NAND

Il bistabile R-S (Reset-Set) realizzato con porte NAND ha i suoi livelli attivi al livello basso, contrariamente a quello precedente. Realizza la stessa funzione di base, però con la logica di ingresso invertita. Questo bistabile ha gli ingressi attivi al livello basso.

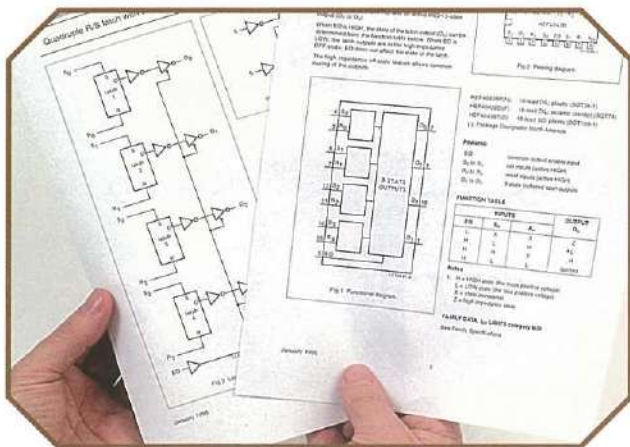
In questo caso, il modo in cui svolge la funzione di memoria elementare è quando mantiene i suoi due ingressi a 1.

Quando l'ingresso /S si attiva a livello basso 0, l'uscita Q passa a 1 (SET) e se l'ingresso /R passa a livello basso 0 l'uscita Q sarà 0 (RESET).

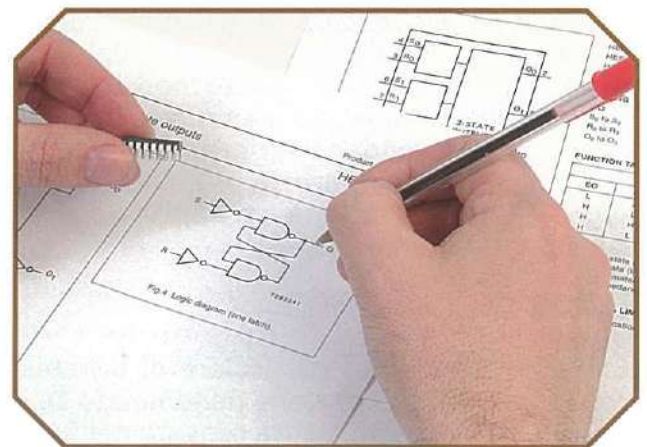
Lo stato che bisogna evitare è quello in cui gli ingressi hanno contemporaneamente il valore 0.



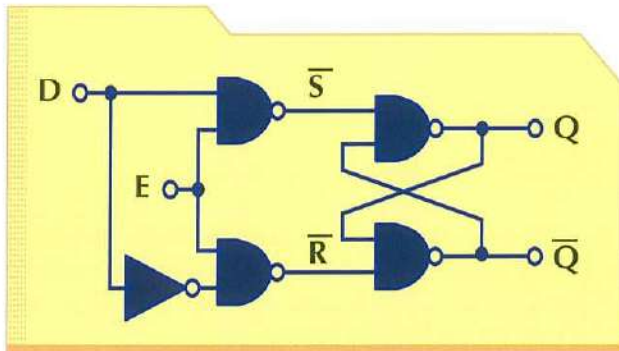
Simbolo del bistabile R-S con ingresso di abilitazione.



Schede tecniche dell'HEF4043B.



Schema logico di uno dei latch del 4043.



D-Latch con ingresso di abilitazione.

E	D	$Q_n$	$/Q_n$	
1	0	0	1	RESET
1	1	1	0	SET
0	X	$Q_{n-1}$	$/Q_{n-1}$	Non cambia

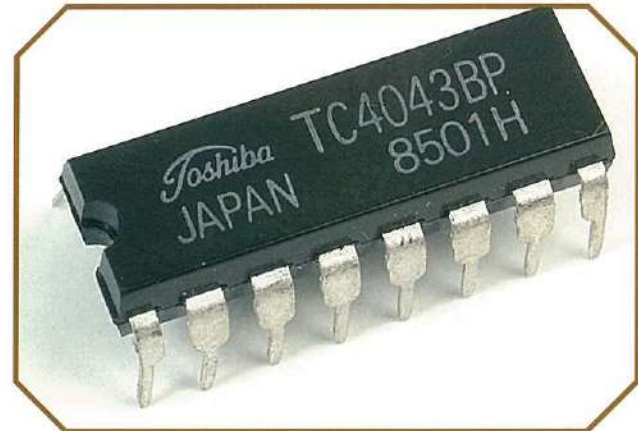
Bistabile D con ingresso di abilitazione.

## Bistabile R-S con ingresso di abilitazione

Alcuni bistabili dispongono di un ingresso addizionale di abilitazione. Il funzionamento è simile a quello del bistabile RS, però gli ingressi sono abilitati solamente quando si applica un 1 sull'ingresso di abilitazione. Normalmente questo terminale è identificato dalla lettera E di Enable. Gli ingressi possono cambiare, però non avranno alcuna influenza fino a quando il terminale E non passa a livello alto. Quando il segnale di abilitazione E è a livello basso, i segnali /S e /R saranno a livello alto indipendentemente dal valore degli ingressi R e S. In questo modo il bistabile è in attesa in modo memoria e garantisce che non avvengano variazioni sull'uscita. Quando si attiva l'ingresso di abilitazione, i segnali R e S si invertono e si applicano al bistabile interno /R e /S.

## Latch D

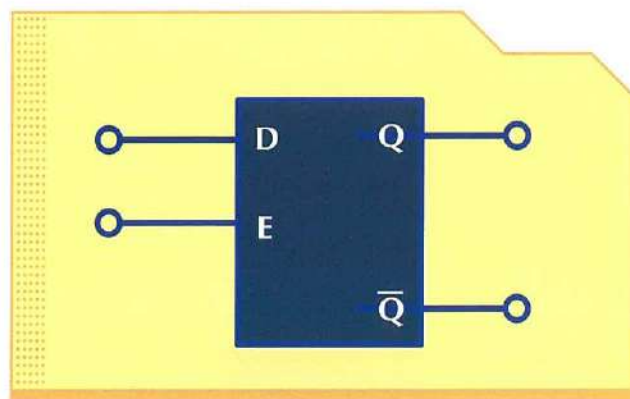
Esiste un tipo molto particolare di bistabile con ingresso di abilitazione denominato D, o D latch. La sua particolarità consiste nel fatto che ha un solo ingresso, indicato con la lettera D, oltre all'ingresso addizionale di abilita-



Latch quadruplo R-S 4043.

zione, che continua a essere indicato con la lettera E. La funzione dell'ingresso di abilitazione è la stessa, quando E è a livello basso i segnali /S e /R del bistabile interno saranno a livello alto e l'uscita del circuito non cambia, dato che si trova in modo memoria.

Quando il terminale E è a livello alto, il valore dell'ingresso D determina i valori dei segnali /R e /S. Quando si applica a D il valore 1, il valore di /S sarà 0 e quello di /R sarà 1, quindi l'uscita passerà a 1, che è lo stato SET del bistabile interno. Quando si applica uno 0 sul terminale D, /S passa a 1, /R passa a 0 e l'uscita Q passa a 0, dato che si forza il RESET del bistabile interno. In realtà questo insieme è un circuito di memoria con abilitazione che contiene un bit di memoria con l'informazione applicata all'ingresso nel momento della sua abilitazione. L'uscita è uguale all'ingresso quando il terminale di abilitazione E è attivo, ovvero, quando si trova a livello 1.



Simbolo del bistabile D.





# Flip-Flop

La denominazione *flip-flop* si applica ai bistabili che hanno l'attivazione per fronte, il quale può essere di salita o di discesa. In un bistabile di questo tipo il cambio di stato sull'uscita si può produrre solamente in modo sincronizzato con il fronte, nel momento in cui quest'ultimo si verifica. Gli ingressi possono cambiare, però conterà solamente il valore che avranno nel momento in cui si genera il fronte.

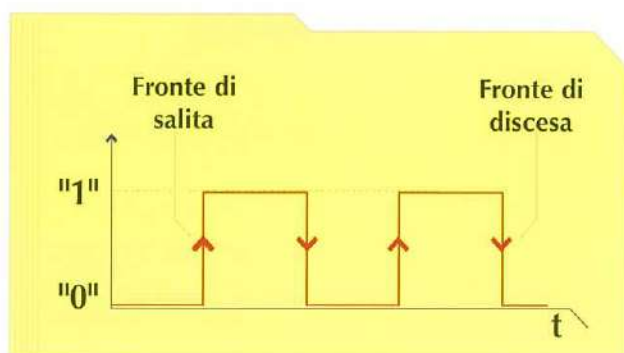
## Il problema

Abbiamo già visto i latch e il loro funzionamento: quando si applicano determinati segnali al loro ingresso si ottiene un valore sull'uscita che si può dedurre dalla tabella della verità, però per ottenere questa variazione sull'uscita ci deve essere una variazione sull'ingresso. L'uscita, quindi, cambia nel momento in cui cambia l'ingresso.

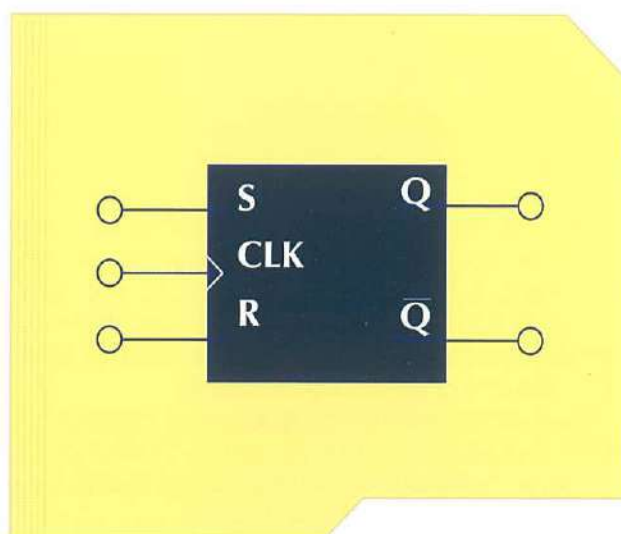
Questo è il funzionamento normale ed è valido per un circuito semplice, però per fare in modo che i circuiti funzionino bene bisogna lavorare in modo "organizzato".

Pensiamo ad esempio a un registro di un computer, la word memorizzata deve rimanere su questo registro il tempo sufficiente per essere letta da un altro circuito del sistema, altrimenti il computer sbaglierà, semplicemente perché questa informazione è arrivata troppo presto o troppo tardi per essere letta da un altro circuito.

Per questo motivo si ricorre ai circuiti sincroni, ovvero a circuiti che pur mantenendo lo stesso funzionamento, possono essere controllati tramite un segnale di clock, però prima spieghiamo che cosa sono i fronti degli impulsi.



Segnale di clock e suoi fronti.

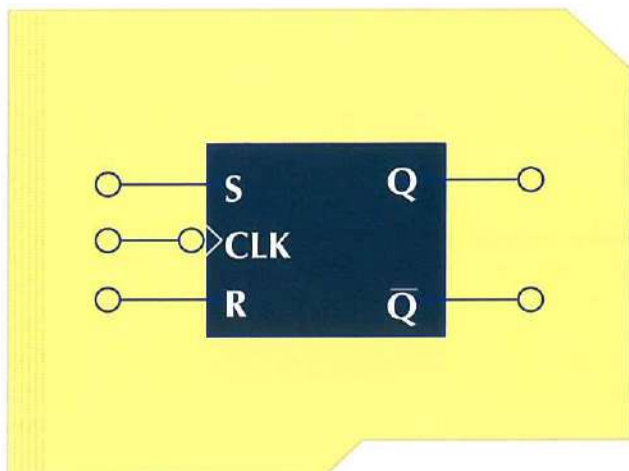


S	R	CLOCK	Q <sub>n</sub>	/Q <sub>n</sub>	
0	0	↑	Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	0	0	Proibito
X	X	X	Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia

Bistabile R-S attivato da fronte di discesa.

## Fronti

Alcuni lettori probabilmente avranno già familiarizzato con la parola fronte, altri no, in ogni caso si tratta di un concetto molto facile da capire. Supponiamo di osservare un segnale di clock, si tratta normalmente di un segnale periodico formato da un treno di impulsi, quando c'è un impulso vi è un livello di tensione ben preciso, ad esempio 5 V, a cui si assegna il valore 1 logico, in assenza dell'impulso abbiamo 0 V, livello che si assegna allo 0 logico.



Bistabile R-S attivato da fronte di salita.

I fronti sono le parti laterali dell'impulso, o i tratti verticali, ovvero, nel momento in cui il segnale passa da 0 a 1 questo tratto prende il nome di fronte di salita o fianco ascendente, mentre quando passa da 1 a 0 si chiama fronte di discesa o fianco discendente.

Negli schemi e nei diagrammi si rappresentano tramite una freccia con la punta verso l'alto per il fronte di salita e con la punta verso il basso per il fronte di discesa.

## Flip-flop

La denominazione flip-flop è largamente utilizzata nella letteratura tecnica anglosassone, invece nei paesi di lingua latina esistono due scuole di pensiero, una utilizza preferibilmente la parola flip-flop, l'altra utilizza genericamente

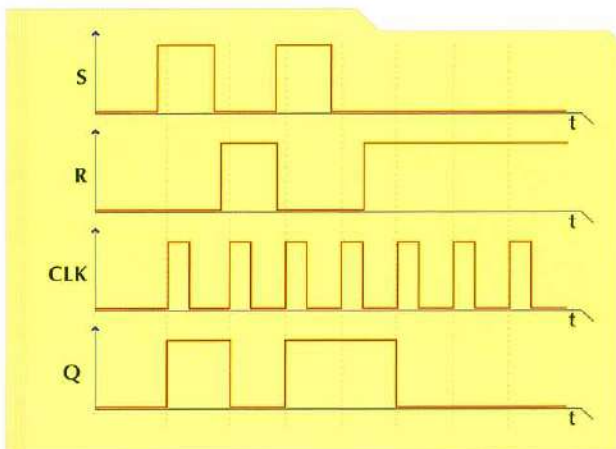
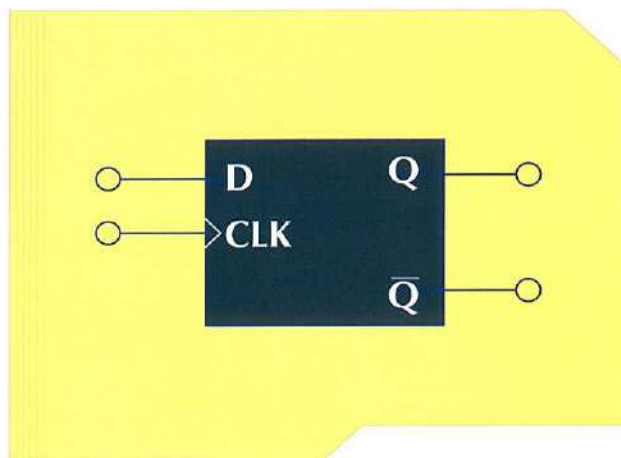


Diagramma dei tempi di un bistabile R-S attivato da fronte di salita.

mente il termine bistabile. Ciò che in realtà deve essere molto chiaro è che si tratta di dispositivi sincroni, ovvero dispositivi che si integrano in un sistema complesso per funzionare in modo sincronizzato con altri, in modo che sia possibile controllare con certezza in quale momento deve cambiare lo stato delle uscite; normalmente il segnale di attivazione è il fronte di salita o di discesa di un impulso. In generale, si tratta di un segnale di clock, anche se può essere un altro tipo di segnale. Una delle possibili applicazioni dei bistabili attivati tramite un fronte è la realizzazione dei registri, con l'ingresso del clock controlliamo l'istante in cui si desidera che la word digitale applicata all'ingresso passi all'uscita per essere processata da un altro circuito.

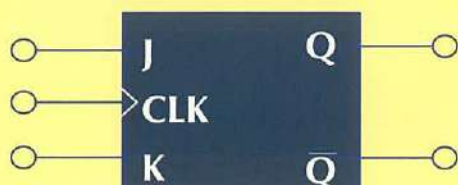
## Simboli

Nei simboli dei bistabili l'ingresso del clock viene indicato con un piccolo triangolo, in questo modo si segnala che l'attivazione si produce sul fronte di salita; però, quando all'esterno del simbolo del componente e di fronte al triangolo che segnala l'ingresso del clock, si aggiunge un piccolo cerchio, questo significa che l'attivazione si produce tramite un fronte di discesa. In altri casi si utilizza il simbolo del fronte di salita e quello di discesa.



Flip-flop tipo D attivato da fronte di salita.

D	CLOCK	Q <sub>n</sub>	/Q <sub>n</sub>	
0	↑	0	1	RESET
1	↑	1	0	SET
X	X	Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia

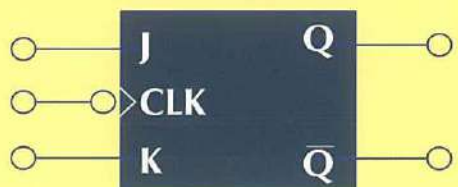


J	K	CLOCK	Q <sub>n</sub>	/Q <sub>n</sub>	
0	0		Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia
0	1		0	1	RESET
1	0		1	0	SET
1	1		/Q <sub>n-1</sub>	Q <sub>n-1</sub>	Cambia

Bistabile tipo J-K attivato da fronte di salita.

### Salita o discesa

Il funzionamento del flip-flop è fondamentalmente lo stesso sia quando lo si attiva con un fronte di salita che con uno di discesa, ma durante la realizzazione del progetto è molto importante tener presente se l'attivazione avviene con il fronte di salita dell'impulso del clock o con quello di discesa, dato che può influenzare in maniera determinante il funzionamento del circuito. Esistono molti tipi di flip-flop integrati e sono disponibili sia con l'attivazione su fronte di salita che su quello di discesa.



Bistabile tipo J-K attivato da fronte di discesa.



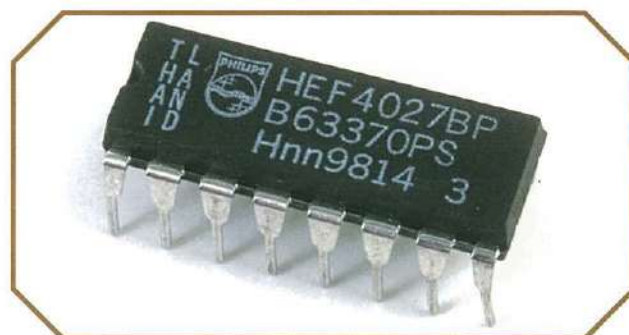
Bistabile tipo T attivato da fronte di salita.

### Flip-flop S-R attivato tramite un fronte

Il funzionamento di questo tipo di bistabile è molto simile a quello del latch R-S, però il circuito legge gli ingressi solamente sul fronte di salita o di discesa del segnale di clock. L'uscita può cambiare solo quando il fronte del segnale che si utilizza per l'attivazione arriva all'ingresso del clock (CLK). Non bisogna confondere questo ingresso con quello di abilitazione (E) dei latch, dato che l'ingresso di abilitazione risponde a un livello di tensione, mentre l'ingresso del clock di un flip-flop risponde a un fronte di un segnale.

Quando non ci sono impulsi di clock, o più esattamente, quando non ci sono fronti o transizioni di clock, il flip-flop rimane nel suo modo memoria e la sua uscita non cambia.

Il funzionamento di un flip-flop R-S attivato tramite un fronte di discesa è identico, eccetto per l'attivazione che avviene sul fronte di disce-



Doppio bistabile tipo J-K MS.



*Doppio bistabile tipo J-K attivato da fronte di discesa.*

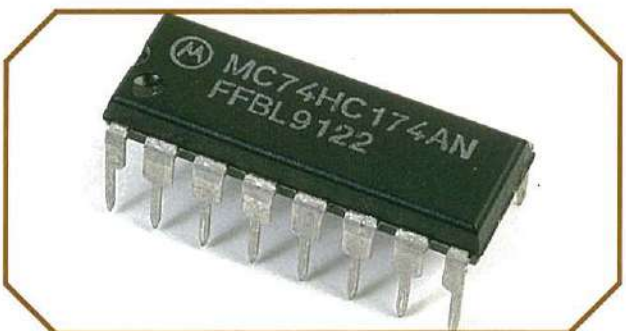
sa del segnale del clock, ovvero quando questo passa dal livello basso al livello alto, in altre parole, quando si produce una variazione da 1 a 0.

### Flip-flop tipo D attivato da un fronte

L'uscita di questo circuito non può cambiare fino a quando non arriva un fronte del clock. Nel momento in cui arriva il fronte del clock il circuito legge l'ingresso e trasferisce il valore sull'uscita. Quindi, per fare in modo che l'uscita cambi è necessario avere un livello diverso sull'ingresso nel momento in cui arriva un fronte del clock; se non cambia l'ingresso, l'uscita non cambia anche se arrivano impulsi di attivazione sull'ingresso del clock.

### Flip-flop tipo J-K attivato da un fronte

Il flip-flop J-K è il più universale di tutti, dato che con esso si possono costruire gli altri tipi. Si comporta come il flip-flop R-S però non ha uno stato di uscita indeterminato, o proibito, come succedeva nei tipi R-S. Se facciamo la comparazione con un bistabile R-S, l'ingresso J equivale all'ingresso S, e l'ingresso K all'ingresso R.



*Sei bistabili tipo D nello stesso integrato.*

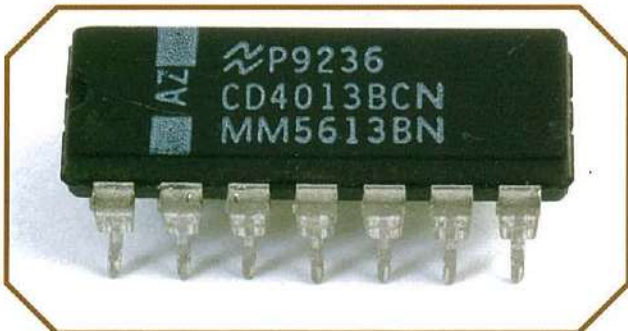


*Doppio bistabile tipo D attivato da fronte di salita.*

Quando i due ingressi sono simultaneamente a livello alto, l'uscita passa allo stato opposto al precedente. Questo tipo di funzionamento è chiamato "basculante". In altre parole, se i due ingressi sono mantenuti a 1, l'uscita cambia ogni volta che all'ingresso del clock si riceve un impulso di clock. Al contrario, se gli ingressi restano a zero l'uscita non cambia, ovvero stiamo lavorando in modo memoria.

### Flip-flop tipo T

Questo tipo di dispositivo ha un ingresso unico indicato con la lettera T. Il suo funzionamento è uguale a quello di un bistabile J-K con i due ingressi uniti. In questo modo, se all'ingresso è stato applicato un livello basso, stiamo lavorando in modo memoria e l'uscita non cambia anche se vengono ricevuti impulsi di attivazione sull'ingresso del clock. Se l'ingresso T passa a livello alto il dispositivo cambia stato ogni volta che si riceve un impulso di attivazione, ovvero, il dispositivo oscilla. Ecco perché in alcuni casi è utilizzata la denominazione oscillatore invece che quella di bistabile.



*Doppio bistabile tipo D attivato da fronte di salita.*



## Il classico 555

**Q**uesto circuito, il cui progetto originale ha più di 30 anni, è tuttora un classico nell'elettronica, e lo utilizzano sia i professionisti che gli appassionati nei diversi settori dell'elettronica. Fu presentato nel 1971 da Signetis Corporation con la denominazione SE555 e NE555, quest'ultimo utilizzato per i modelli commerciali di uso corrente. È stato ed è tuttora prodotto da molte case, il suo progetto è stato modificato e migliorato nel corso degli anni, però questi cambiamenti hanno conservato la compatibilità della piedinatura. Ne esistono anche delle versioni a basso consumo realizzate con tecnologia CMOS il cui funzionamento è simile anche se lo schema interno è diverso, a causa della tecnologia utilizzata.

### Vantaggi

Questo circuito battezzato come "timer", che in italiano si potrebbe tradurre come "temporizzatore", ha le sue principali applicazioni nei progetti di circuiti monostabili e astabili.

I monostabili sono circuiti la cui uscita permane un certo tempo in un determinato stato dopo l'attivazione e, trascorso il tempo prefissato, torna allo stato originale.

Gli astabili sono circuiti il cui segnale di uscita è periodico sotto forma di impulsi, si tratta di oscillatori la cui uscita è un'onda quadra.

Per capire il funzionamento del 555 si utilizza un diagramma a blocchi molto semplificato, che però rappresenta esattamente il suo funzionamento. Per descrivere il funzionamento di questo circuito faremo riferimento al contenitore più comune, il DIL-8.

Nella tabella possiamo vedere la nomenclatura di ogni terminale e anche la denominazione più utilizzata, sia in italiano che in inglese, dato che sono utilizzate entrambe indistintamente.

Osservando il diagramma a blocchi possiamo notare subito due amplificatori operazionali, AO1 e AO2, una rete da tre resistenze di valore uguale, un bistabile R-S, due transistor

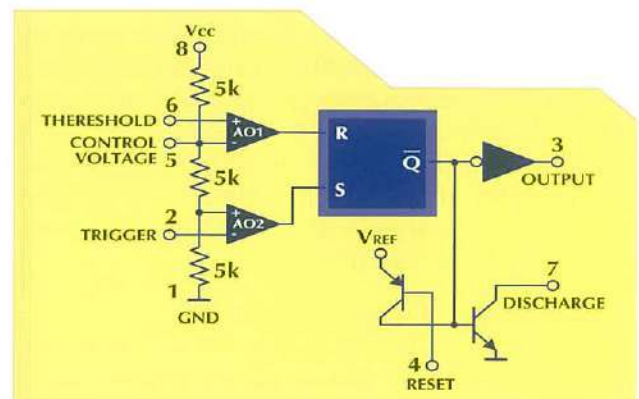


Timer 555.

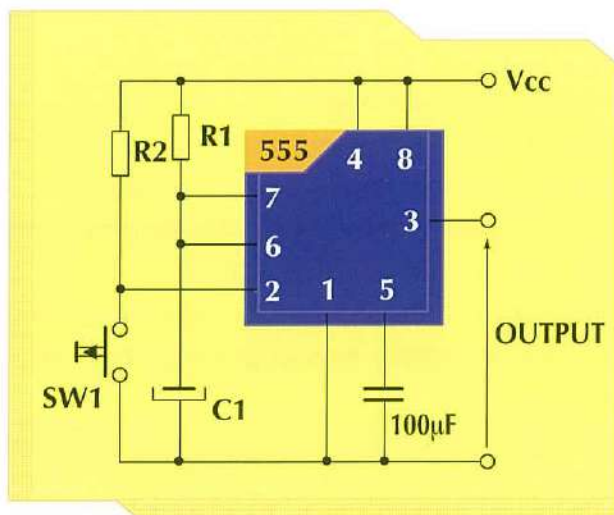
Terminale	Descrizione	
1	Massa	Ground (GND)
2	Attivazione	Trigger
3	Output	Uscita
4	Reset	Reset
5	Tensione di controllo	Controllo tensione
6	Soglia	Threshold
7	Scarica	Discharge
8	Vcc	Vcc

e un buffer invertente che configura l'uscita del circuito. Le tre resistenze di valore uguale, normalmente da 5 K, sono collegate tra Vcc, ovvero il positivo dell'alimentazione e la massa (GND), e sono utilizzate per ottenere due tensioni che servono da riferimento per gli amplificatori operazionali: 1/3 di Vcc come riferimento per l'ingresso non invertente di AO2 e 2/3 di Vcc come riferimento per l'ingresso invertente di AO1.

Per quanto riguarda i comparatori abbiamo due possibili livelli di tensione di uscita, 0 V e Vcc, che assegniamo ai valori logici 0 e 1 rispettivamente.



Schema a blocchi dell'interno del 555.



Configurazione tipica come monostabile.

Quando la tensione sul terminale "soglia", che è l'ingresso non invertente di AO1 (+), supera i 2/3 di Vcc, la sua uscita passa al livello logico 1; questa uscita è collegata all'ingresso R del bistabile e quindi la sua uscita negata /Q passa al valore logico 1, l'uscita passa al livello 1 provocando anche la saturazione del transistor Q1, portando quindi il terminale 7 a livello basso.

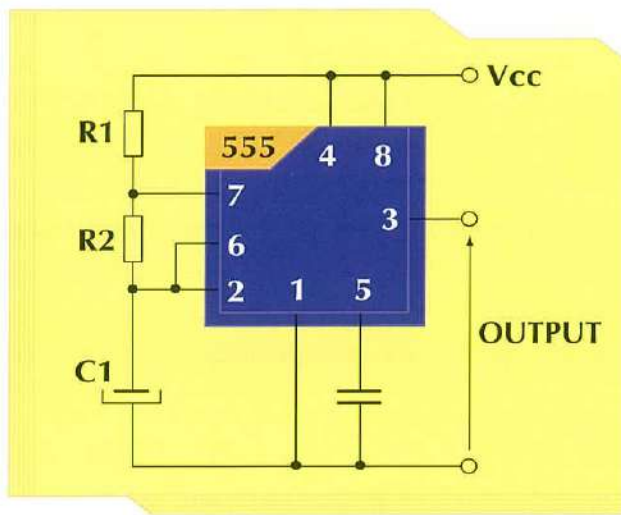
Quando, però, la tensione sul terminale "attivazione", che corrisponde all'ingresso invertente di AO2 (-), scende sotto 1/3 di Vcc, la sua uscita passa a 1, essendo collegata all'ingresso S del bistabile, quest'ultimo passa a 1, quindi l'uscita del bistabile /Q passa a 0, l'uscita passa al livello alto 1 logico e il transistor Q1 entra in stato di interdizione (non conduce).

### Reset

Il terminale di reset, pin 4 sul tipo di contenitore più comune, funziona quando si imposta a zero. Quando questo pin passa a zero si satura il transistor Q2, l'uscita passa a zero e si pone in interdizione il transistor Q1. In molte applicazioni questo terminale non si utilizza, in questo caso si collega direttamente il positivo dell'alimentazione Vcc.

### Alimentazione

Questo circuito si alimenta tra i terminali 8 (Vcc) e 1 (GND). La tensione applicata al terminale 8 deve essere compresa tra 4,5 e 15 V,



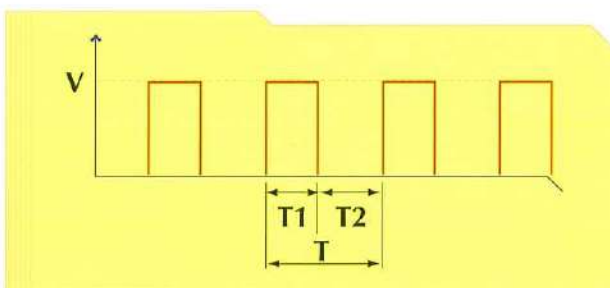
Configurazione tipica come astabile.

limite che non deve essere superato, anche se i costruttori accettano fino a 16 V e, per le versioni dedicate alle applicazioni militari, fino a 18 V.

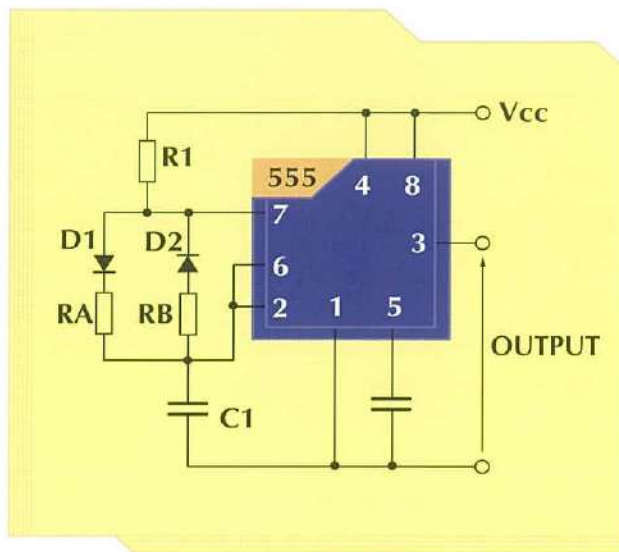
Normalmente viene alimentato con la stessa tensione con cui lavora il circuito a cui è associato il componente. Ad esempio, se si utilizza con integrati della famiglia TTL si alimenta a 5 V e i suoi livelli di uscita sono compatibili con quelli di questa famiglia. Quando si utilizza la famiglia CMOS 4000 si può lavorare a 5, 9 o 12 V, secondo la convenienza. Il consumo di questo circuito dipende dai componenti esterni e va da un minimo di 3 mA alimentato a 5 V fino a un massimo di 10 mA alimentato a 15 V.

### Uscita

Uno dei principali vantaggi di questo circuito integrato è la sua capacità di gestire corrente all'uscita, dato che può assorbire o erogare fino a 200 mA, una corrente molto elevata che

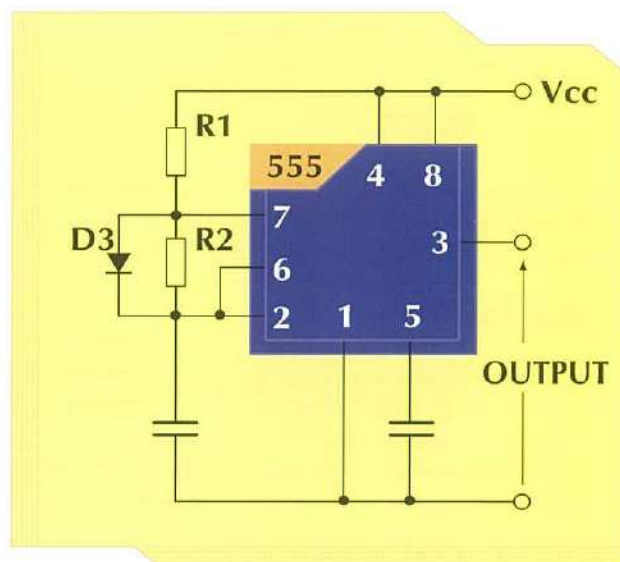


Chiamiamo T1 la durata dell'impulso, T è il periodo.



Astabile con regolazione indipendente di T1 e T2.

permette di eccitare direttamente molti circuiti o dispositivi, compresi quelli elettromeccanici come i relé o le valvole ad azionamento elettromagnetico, senza la necessità di utilizzare componenti esterni in moltissime applicazioni. Questo integrato può dissipare fino a 600 mW.



Astabile che genera impulsi di breve durata.

### Monostabile

L'obiettivo di questo circuito è mantenere l'uscita a un livello alto per un determinato tempo a partire dal momento in cui si attiva. Quando il circuito è a riposo l'uscita dello stesso, il terminale 3, rimane a livello basso. Premendo per un attimo il pulsante di attivazione si collega a massa il terminale 2 "attivazione". A partire da questo momento l'uscita del circuito passa a livello alto e rimane a questo livello il tempo fissato dal valore della resistenza R1 e dalla capacità del condensatore C1. Se guardiamo lo schema si può notare che, oltre al circuito integrato, ci sono pochi componenti esterni. Gli altri componenti fondamentali del circuito sono la resistenza R1 e il condensatore C1. Per calcolare la temporizzazione, ovvero il tempo

EQUIVALENT SCHEMATIC

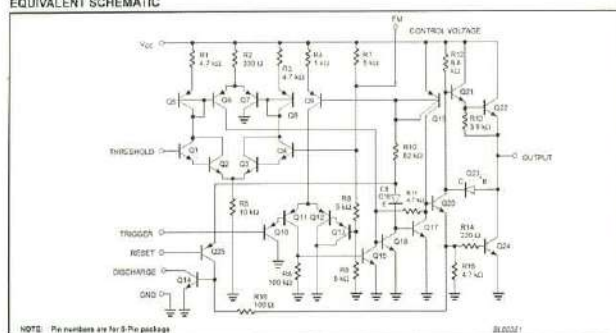
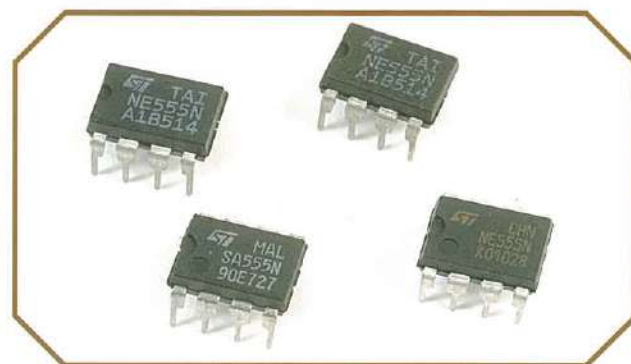


Figure 3. Equivalent schematic

Schema elettrico presente sul catalogo del costruttore.

### Applicazioni

Descriveremo ora le due applicazioni più utilizzate e alcune delle loro varianti più interessanti. L'applicazione come monostabile, come bistabile e la possibilità di controllare in modo indipendente la durata del livello alto del periodo T1 e la durata del livello basso di tensione dello stesso T2; il periodo è la somma di entrambi i tempi:  $T = T1 + T2$ .



555 di diverse fabbriche dello stesso costruttore.



in cui l'uscita del circuito rimane a livello alto, si utilizza la seguente formula matematica:

$$T = 1,1 \times R1 \times C1$$

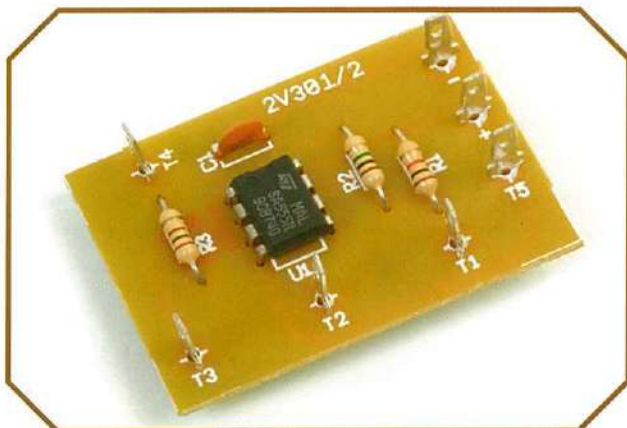
Il valore della resistenza R1 deve essere espresso in Ohm e quello della capacità C1 in Farad. Il risultato si ottiene in secondi.

### Precisione

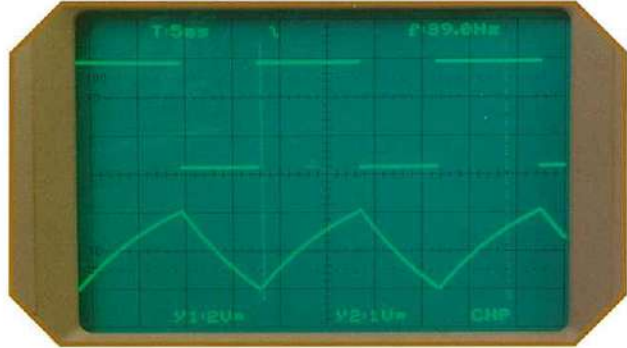
La precisione del circuito è dell'1%, normalmente però si utilizzano valori di resistenze con una precisione del 5% e condensatori del 10%. Il valore dei condensatori elettrolitici di utilizzo corrente può avere una tolleranza che va da -20% a +50%. Inoltre le formule sono approssimate. Tuttavia la pratica dimostra che sono molto utili, e quando si richiede precisione bisogna inserire nel circuito alcuni elementi di regolazione, ad esempio un potenziometro. Inoltre bisogna sempre verificare il funzionamento del circuito e misurare i tempi con uno strumento adeguato, non dobbiamo fidarci ciecamente dei calcoli né dei componenti, è necessario verificare sempre il funzionamento dopo aver terminato.

### Astabile

L'uscita del circuito astabile è un segnale periodico sotto forma di impulsi. L'impulso ha una durata T1 e la parte del periodo in cui l'uscita rimane a livello basso, ovvero quando non c'è uscita, si identifica come T2. Il periodo del segnale è, quindi, la somma di entrambi i



Circuito reale che utilizza il 555.



Andamento della tensione di uscita e di quella del condensatore.

tempi. I tempi dell'astabile si calcolano utilizzando le seguenti formule:

$$T1 = 0,7 (R1 + R2) C1$$

$$T2 = 0,7 R2 C1$$

Non bisogna dimenticare che le resistenze si esprimono in Ohm, la capacità in Farad e il risultato si ottiene in secondi.

### Astabile con controllo di T1 e T2

Questo circuito è un astabile con una piccola modifica che comprende i diodi D1 e D2, per rendere indipendente il percorso di carica e scarica del condensatore. La carica si realizza tramite R1 e RA e la scarica tramite RB; in questo caso bisogna modificare le formule:

$$T1 = 0,7 (R1 + RA) C1$$

$$T2 = 0,7 RB C1$$

Questo circuito si utilizza normalmente con elementi di regolazione associati alle resistenze RA e RB, per modificare il valore delle stesse e per ottenere dei tempi molto precisi dopo la regolazione.

### Astabile con impulsi di breve durata

Questo circuito si utilizza quando sono necessari impulsi molto brevi e la durata degli stessi è controllata dalla resistenza R1, che può essere di basso valore. Le formule da applicare in questo caso sono:

$$T1 = 0,7 R1 C1$$

$$T2 = 0,7 R2 C1$$





## MPLAB

**M**PLAB è uno strumento software per il progetto e lo sviluppo di applicazioni con microcontroller PIC della ditta Arizona Microchip Technology (AMT). Contiene tutte le utilità necessarie per la realizzazione di qualsiasi progetto, permette la scrittura del file sorgente in linguaggio assembler, la compilazione e la simulazione su display. Inoltre permette l'esecuzione del programma in modalità passo a passo per vedere come evolvono in modo reale i registri interni, la memoria RAM e/o EEPROM e la memoria di programma, mano a mano che si eseguono le istruzioni.

### Installazione del programma

Il CD allegato contiene la versione 5.70 di MPLAB. Se carichiamo il CD nel nostro lettore e selezioniamo l'opzione "Programmi", potremo vedere apparire l'icona di installazione mostrata nella figura.



Icona di installazione di MPLAB.

Quando eseguiamo il file di installazione appare una videata di benvenuto che ci chiede se desideriamo installare l'applicazione.

Se accettiamo con il pulsante Next il programma risponde con una nuova videata dove dovremo accettare le condizioni del contratto di licenza per l'installazione. Se accettiamo le condizioni "I agree" e clicchiamo nuovamente su Next, apparirà un'altra videata in cui potremo scegliere i moduli dell'applicazione che desideriamo installare.

Dato che lo spazio richiesto per installare l'applicazione completa non è molto grande, lasceremo selezionati i moduli che appaiono per default e cliccheremo Next in questa videata e in quella successiva.

È possibile cambiare la directory in cui viene installata l'applicazione anche se consigliamo di utilizzare quella che appare per default (C:\Programmi\MPLAB). Per completare l'installazione bisognerà selezionare Next nelle successive videate e, infine, riavviare il computer.

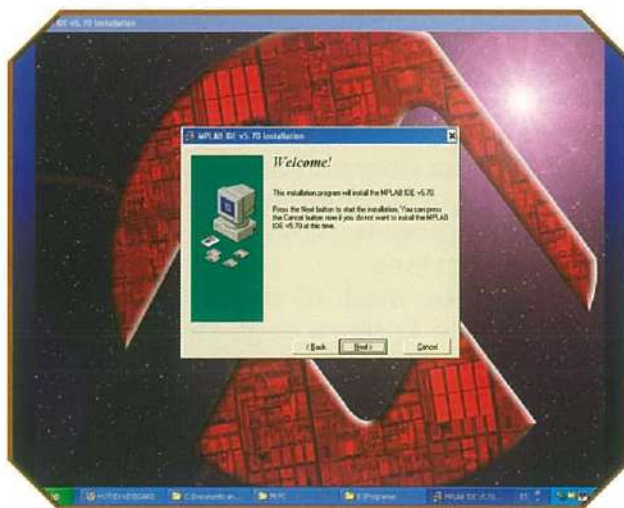
Terminata l'installazione, dato che utilizzeremo molto spesso questo programma, vi consigliamo di creare un'icona di accesso diretto

sul desktop. Vi consigliamo inoltre di creare una directory per memorizzare i vostri programmi e i file generati da questi, ad esempio: C:\Programmi\MPLAB\Progetti.

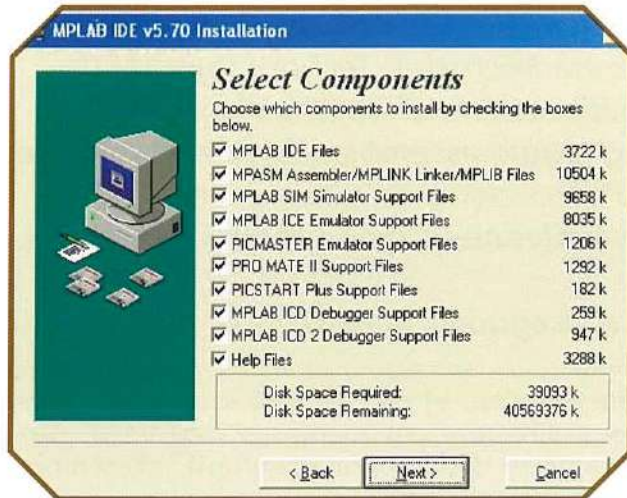
### Configurazione di MPLAB

A questo punto siamo già pronti per utilizzare il programma. Cliccando sull'icona di MPLAB apparirà una videata come quella riportata nella figura della pagina successiva. Dobbiamo selezionare come intendiamo lavorare su MPLAB e quale microcontroller utilizzeremo. A questo scopo selezioneremo Option dal menù superiore e, all'interno del sottomenù a tendina, selezioneremo Development mode. Qui dovremo selezionare MPLAB SIM Simulator e il processore da utilizzare, che nel nostro caso è il PIC16F870.

Se selezioniamo la cartella Clock possiamo scegliere l'oscillatore da utilizzare. Lavoreremo con un oscillatore di tipo XT a una fre-



Videata iniziale dell'installazione di MPLAB.



*Moduli di MPLAB che si possono installare.*

quenza di lavoro da 4MHz. Applicheremo le variazioni con Apply e usciremo dalla configurazione accettando i cambiamenti con OK. Nella barra inferiore del programma potremo osservare le nostre modifiche.

## La barra degli strumenti



*Icona di accesso diretto.*

MPLAB fornisce accessi rapidi alle principali funzioni mediante delle icone poste sulla barra degli strumenti. È possibile commutare quattro barre di funzioni diverse a seconda di ciò che desideriamo fare con il programma.

## Inizio della programmazione

Quando vorremo sviluppare un programma, per prima cosa dovremo includerlo in un progetto. Il progetto comprenderà sia il file in assembler che progetteremo, che i file risultanti dalla compilazione.

Esistono due modi di creare un progetto, uno mediante la barra degli strumenti dove potremo scegliere l'icona adeguata, l'altro tramite i menù, selezionando File e, all'interno di questo, New. In entrambi i modi apparirà il riquadro di dialogo "Create project" mostrato nella figura.

Se rispondiamo Yes alla domanda che ci viene posta, apparirà una finestra in cui potremo

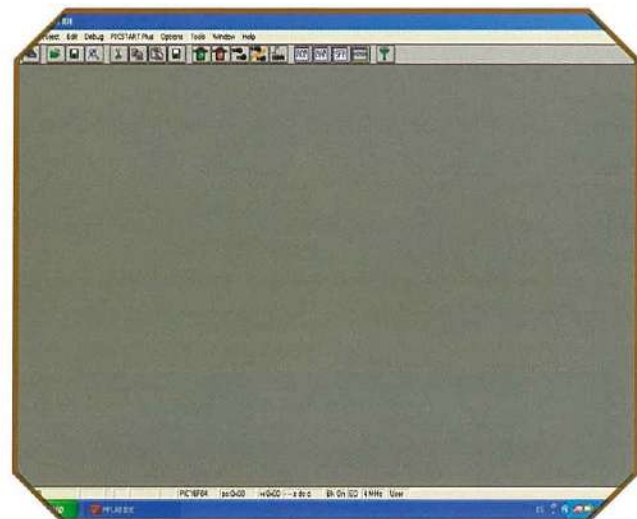


*Installeremo l'applicazione nella directory che ci viene proposta per default.*

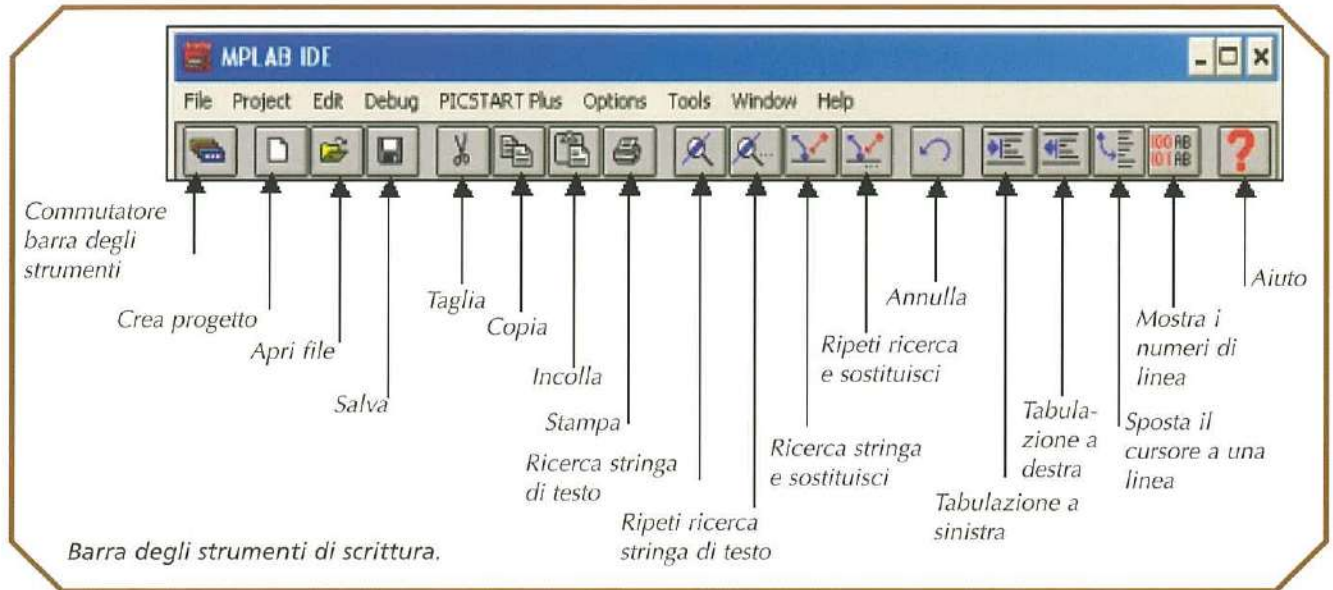
assegnare un nome al progetto e scegliere la directory dove memorizzare quest'ultimo. L'indirizzo che utilizzeremo normalmente sarà: C:\Programmi\MPLAB\Progetti e il nome che daremo potrà essere uno qualsiasi, sempre che rispetti l'estensione .pjt (ad esempio: prova1.pjt).

Se clicchiamo OK apparirà il riquadro di dialogo riportato nella prima figura della pagina successiva.

Cliccando nuovamente OK potremo iniziare a creare il nostro primo programma, infatti che apparirà una videata bianca dove sarà possibile scrivere il file. Nel nostro caso, dato che ancora non sappiamo programmare, l'u-



*Videata iniziale di MPLAB.*



nica cosa da fare sarà copiare in questa finestra un programma progettato in precedenza.

### Carico dell'esempio

Sarà necessario caricare il CD che vi è stato fornito nel lettore del computer e selezionare Esercizi e Applicazioni. Aprire il primo esercizio "ese1.asm", selezionare tutto il testo e copiarlo. In seguito copiatelo sull'editor di MPLAB. La videata avrà l'aspetto della figura relativa.

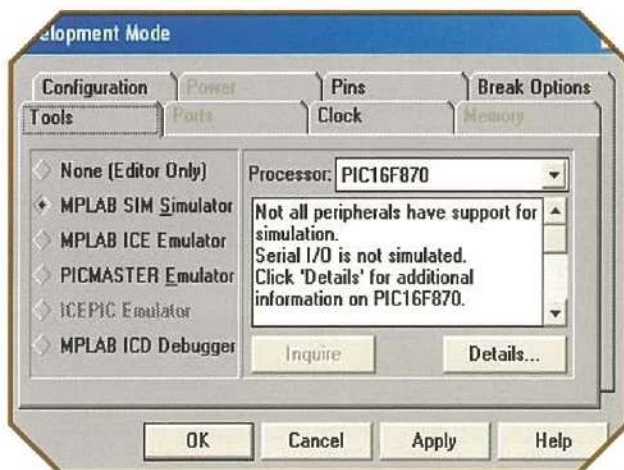
Come potrete verificare la prima colonna dell'editor è riservata alle etichette. Le etichette sono espressioni alfanumeriche gene-

rate dall'utente che si associano a posizioni di memoria, è necessario iniziare da una lettera e non potranno mai essere usate espressioni utilizzate dall'assembler quali istruzioni, direttive del linguaggio, nomi di registri speciali o nomi di bit di questi registri.

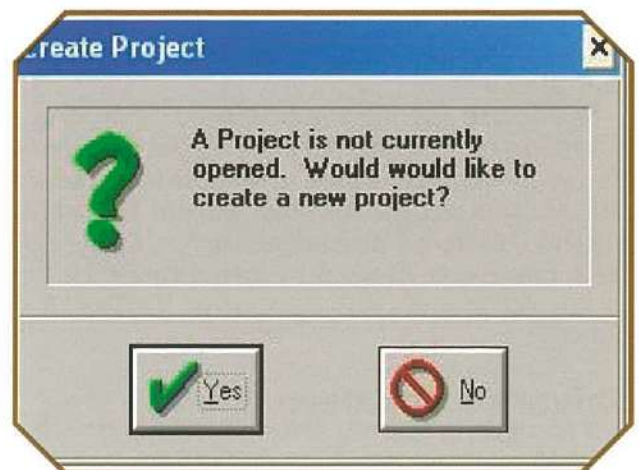
Nella colonna successiva si scrive lo mnemonico delle istruzioni o direttive del linguaggio di programmazione.

### Commenti di testo

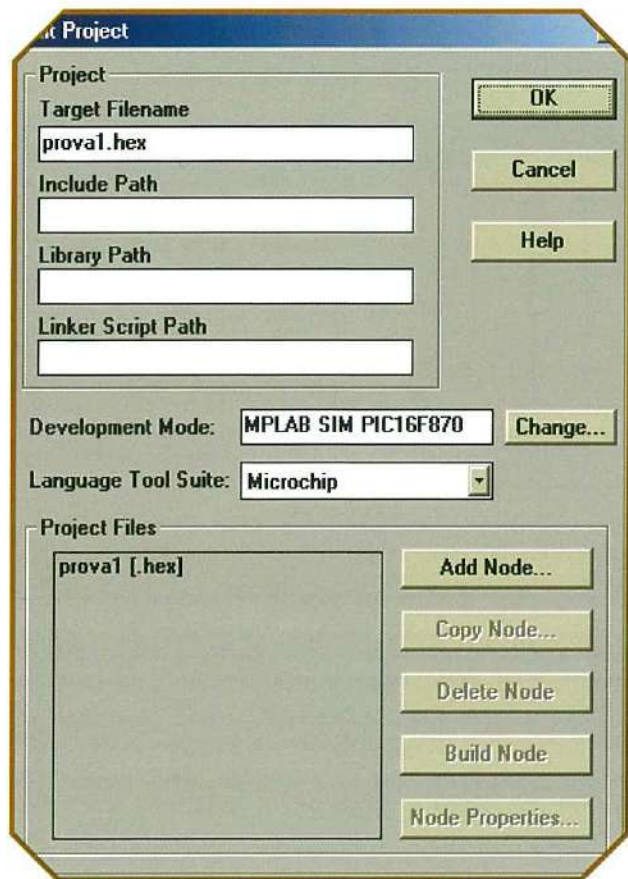
È fondamentale che un programma sia commentato, in altre parole che il programmatore spieghi cosa intende fare su ogni linea di



Configurazione del programma.



Videata per la creazione di un nuovo progetto.



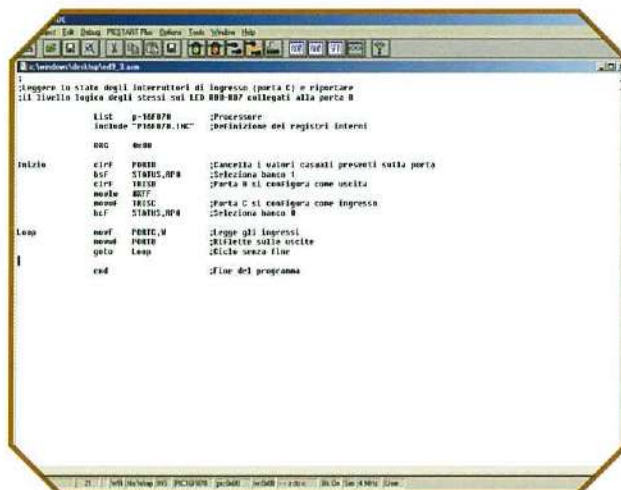
Vedeata in cui si assegnano al progetto i file di scrittura.

programmazione. In questo modo qualsiasi persona potrà capire facilmente il programma e potrà ampliarlo o modificarlo. Per inserire un commento è necessario porre ";" (punto e virgola) davanti al testo. Così facendo, quando il compilatore incontra questo simbolo non genererà codice macchina con ciò che segue sulla stessa linea.

Nel nostro esempio le prime tre linee di codice sono commenti. All'interno dei commenti non contano gli spazi, le linee lasciate bianche o l'utilizzo di qualsiasi parola riservata, purché il tutto sia preceduto da ";". Come potrete osservare, dopo ogni istruzione si utilizza un commento per spiegarne il funzionamento.

## Salvare e chiudere

Quando avremo terminato di editare il codice in assembler (in questo caso avremo già terminato perché abbiamo copiato un programma

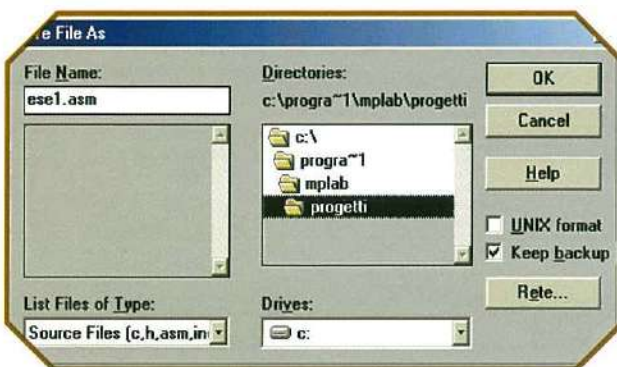


Primo programma con MPLAB.

completo) occorre salvare il file. Per questo selezioneremo File e poi Save. Apparirà il riquadro di dialogo della figura soprastante in cui dovremo assegnare un nome al file creato. Chiameremo il file "prova1.asm", ci assicuriamo di aver selezionato la directory dove intendiamo salvare i nostri progetti e cliccheremo OK.

Ora potremo uscire dal programma. Selezionando la x nell'angolo superiore destro della finestra di programma, con File ed Exit il programma chiederà se vogliamo salvare i cambiamenti nel progetto, domanda alla quale dovremo rispondere Yes prima di uscire dall'applicazione.

Quando apriremo nuovamente il programma, questi ci chiederà se desideriamo aprire l'ultimo progetto memorizzato. Vi consigliamo di selezionare No e ripetere quanto imparato finora con un altro degli esercizi contenuti sul CD.



Salviamo il nostro programma.



## Bistabili M-S

**Q**uando dobbiamo lavorare con diversi circuiti in modo sincronizzato nasce il problema della differenza di velocità di risposta dei circuiti fra loro; questo può causare difficoltà di funzionamento nei circuiti contatori, i quali utilizzano diversi bistabili che lavorano in modo coordinato e che devono cambiare stato contemporaneamente. In questi casi può succedere che alcuni dispositivi cambino prima del tempo e i segnali di uscita assumano valori inadeguati, anche se per breve tempo, procurando gravi problemi di funzionamento, dato che questi segnali potrebbero influenzare altri circuiti.

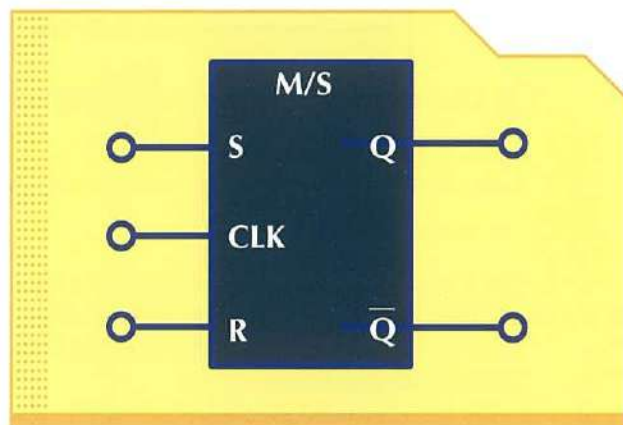
### Bistabili M-S

I bistabili M-S risolvono il problema appena esposto. Fondamentalmente si tratta di due bistabili collegati in serie e con il clock invertito. È molto facile trovare circuiti integrati Master-Slave dei tipi R-S, J-K e D, dato che sono i più utilizzati. Normalmente si usa la denominazione inglese Master Slave.

Possiamo spiegarne il funzionamento in modo molto semplificato tenendo presente che si seguirà la seguente procedura: si parte dall'uscita che viene bloccata, si legge l'ingresso, si memorizza internamente l'ingresso, si blocca l'ingresso, si permette al dato memorizzato di essere elaborato sul bistabile di uscita e quest'ultimo fornirà l'uscita.

### Bistabile R-S M-S

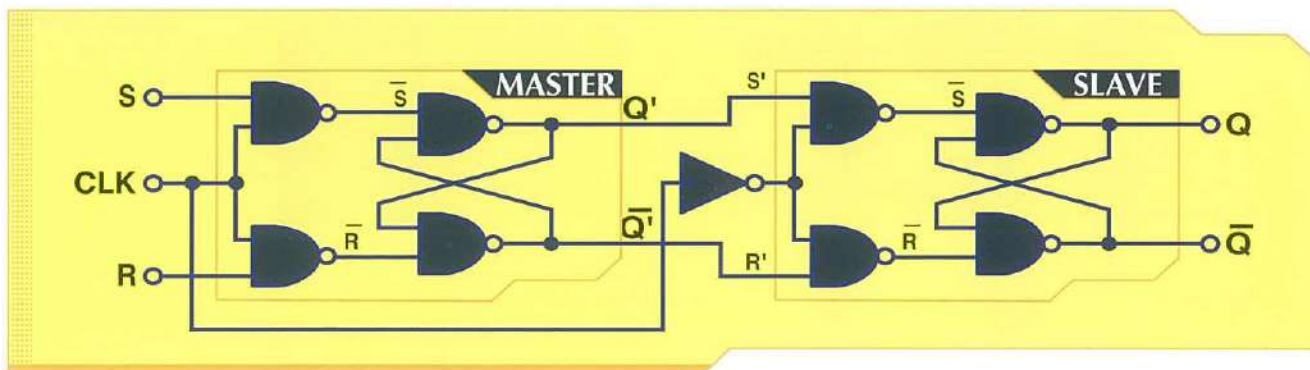
Conoscendo il funzionamento del bistabile R-S risulta semplice capire anche il funzionamento del bistabile R-S Master Slave. Osservando la figura potremo vedere due bistabili R-S collegati in serie. La cosa più interessante è che l'ingresso del clock che si applica al pri-



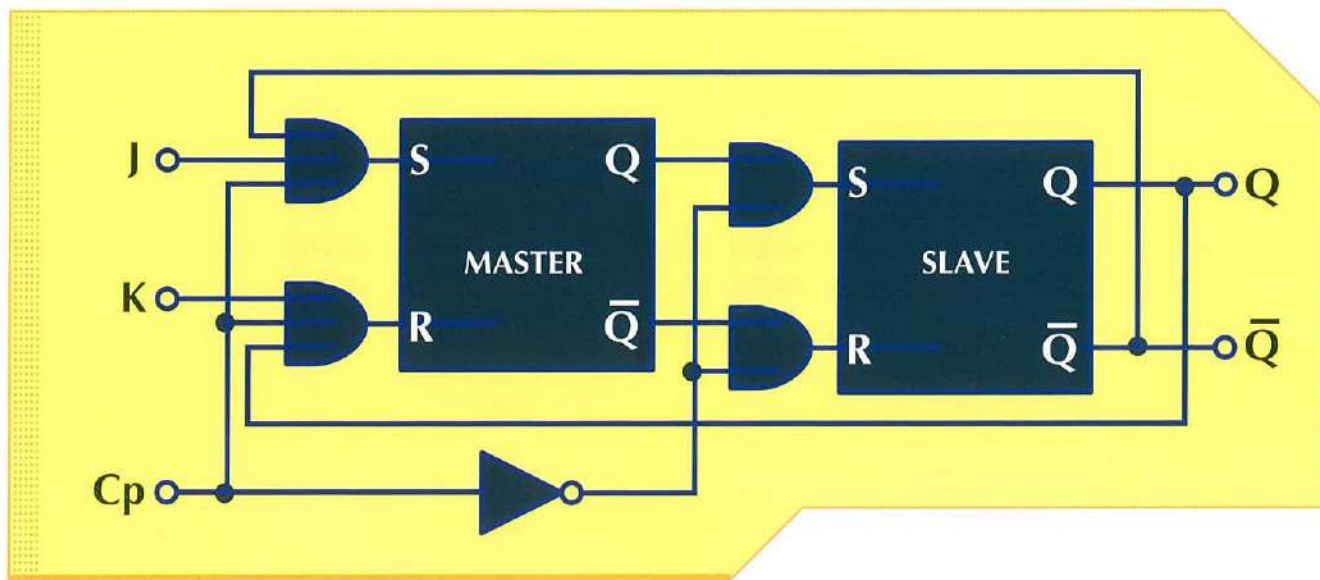
Simbolo di un bistabile R-S Master Slave.

mo bistabile viene invertito prima di essere applicato al secondo.

Vediamo ora, passo a passo, come funziona questo circuito. Quando il segnale di clock applicato al primo bistabile passa a livello alto, si abilita il primo bistabile identificato come Master e contemporaneamente il segnale invertito disabilita il secondo bistabile, ovvero lo Slave, quindi l'uscita in quel momento non può cambiare.



Bistabile R-S M-S (Master Slave).



Bistabile J-K Master Slave.

Quando il segnale di clock diventa zero il bistabile Master viene disabilitato quindi mantiene il suo stato, lo Slave si abilita e legge i dati applicati al suo ingresso.

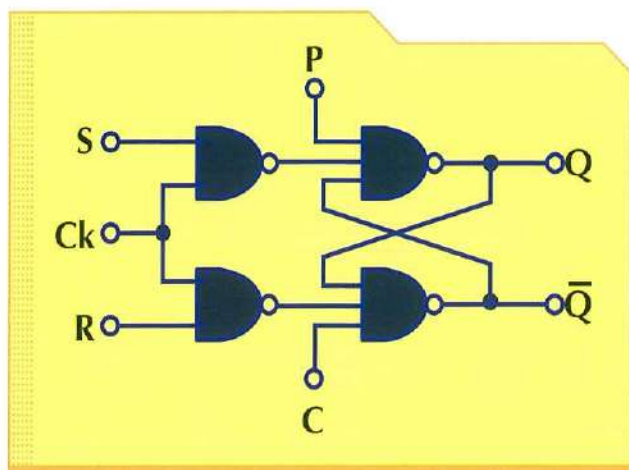
Tornando allo schema possiamo vedere che le uscite Q e /Q del Master sono collegate a S e R rispettivamente dello Slave. In questo modo agli ingressi dello Slave può arrivare solamente 01 oppure 10, dato che sono collegate alle due uscite e sono sempre una l'inverso dell'altra.

Se l'uscita Q del Master è 1 lo Slave sarà sempre in stato di RESET, dato che la Q del Master è collegata all'ingresso S dello Slave. Se la Q è

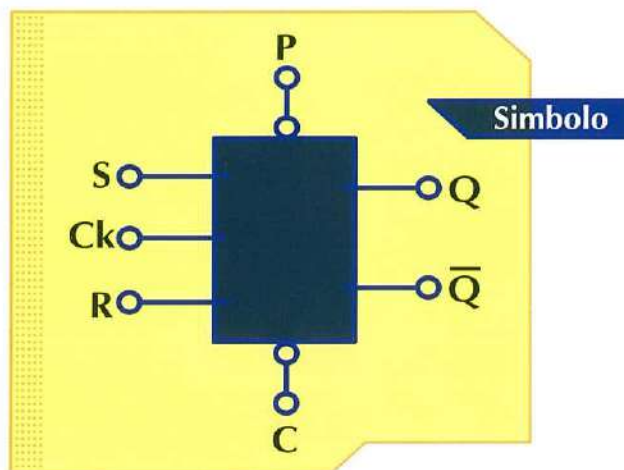
zero verrà applicato un 1 all'ingresso R, quindi lo stato sarà RESET.

Riassumendo, lo Slave prende lo stato di uscita del Master che non può cambiare durante questa lettura, dato che il Master è disabilitato.

La tabella della verità è la stessa di quella di un R-S attivato da un fronte, però bisogna tener presente che deve essere applicato completamente l'impulso di clock, in altre parole il bistabile J-K M-S legge gli ingressi mentre il clock è a livello alto, però l'uscita non è valida fino a quando l'impulso non torna a livello basso.



Bistabile R-S con ingressi asincroni.



Simbolo di un bistabile R-S con ingressi asincroni.



S	R	CLOCK	Q <sub>n</sub>	/Q <sub>n</sub>	
0	0	(*)	Q <sub>n-1</sub>	/Q <sub>n-1</sub>	Non cambia
0	1	(*)	0	1	RESET
1	0	(*)	1	0	SET

(\*) Impulso completo, l'uscita è valida quando il segnale torna a livello basso.

Bistabile R-S M-S.

### Bistabile J-K M-S

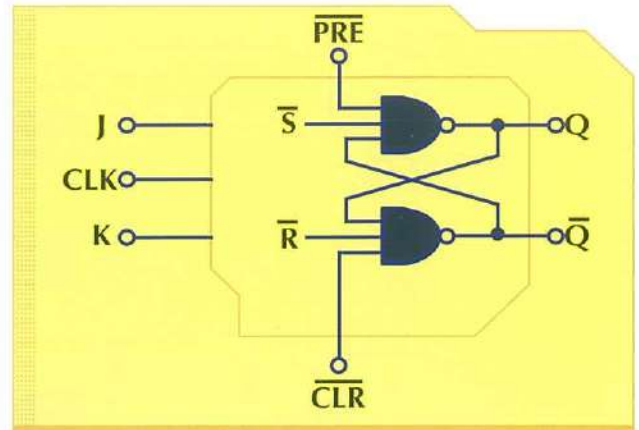
Il bistabile J-K M-S è probabilmente il migliore dal punto di vista dell'utilizzo. Ha un funzionamento molto simile a R-S, se osserviamo lo schema vedremo che la sua principale caratteristica è il disporre di una retroazione di uscita sull'ingresso del Master che, in questo modo, tiene conto dello stato dell'uscita.

Quando il clock è a zero il Master resta disabilitato e non accetta variazioni ai suoi ingressi J-K.

Tuttavia quando il clock passa a 1 il Master può cambiare stato se gli ingressi J-K e lo stato della retroazione lo permettono, senza però influenzare lo Slave che in questo caso è disabilitato.

### Ingressi asincroni ausiliari Preset e Clear

Generalmente nei circuiti si utilizzano dispositivi sincroni, ovvero dispositivi che leggono i loro ingressi sul fronte di salita o di discesa di un impulso di clock, in alcuni casi, però, è necessario forzare lo stato delle uscite a causa di

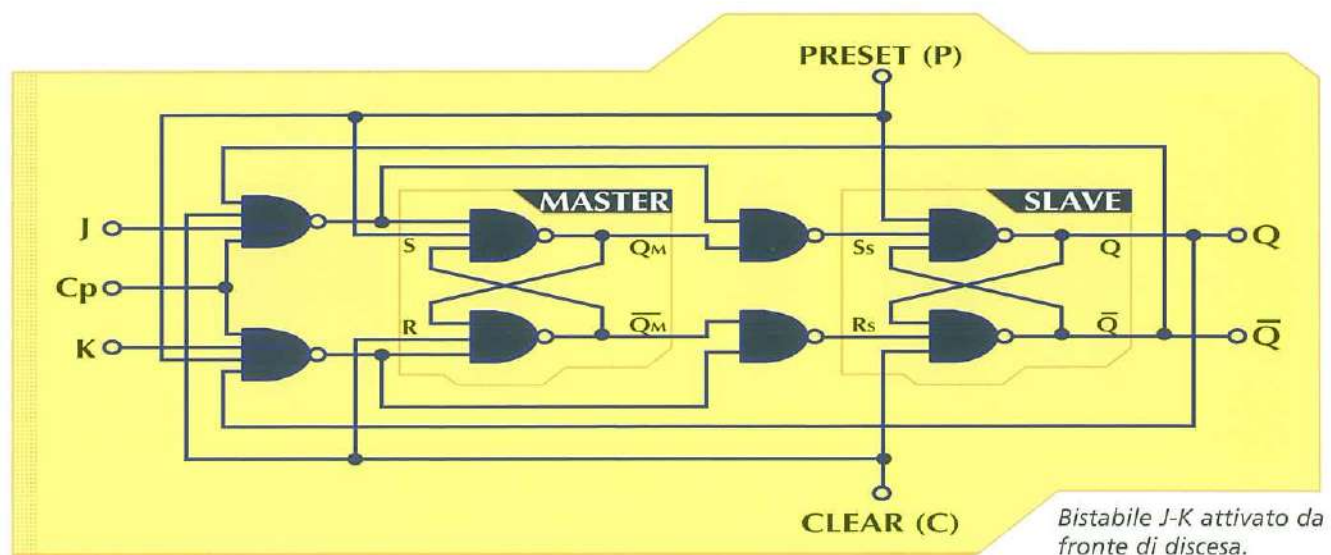


Dettaglio degli ingressi asincroni di un bistabile J-K.

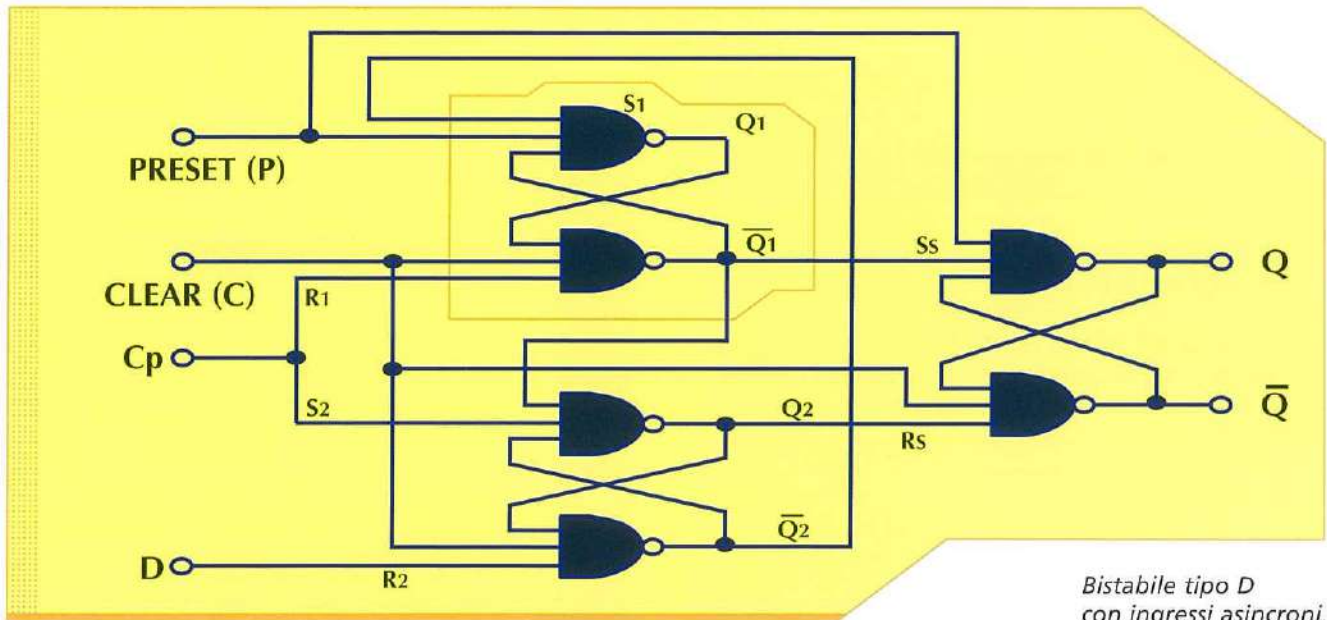
altri fattori esterni, pur mantenendo il normale funzionamento in assenza di questa necessità. Per risolvere questo problema si parte da un bistabile sincrono a cui si aggiunge una circuiteria aggiuntiva che permette di sommare degli ingressi di controllo ausiliario denominati PRESET e CLEAR, i quali attivandosi servono per l'inizializzazione del bistabile PRESET, oppure la cancellazione o l'impostazione a zero, CLEAR. Non tutti i bistabili hanno questi ingressi.

### Clear

L'ingresso asincrono non funziona fino a quando non è attivato. Quando si desidera impostare l'uscita del bistabile a zero si attiva l'ingresso di controllo CLEAR. Questo ingresso



Bistabile J-K attivato da fronte di discesa.



Bistabile tipo D  
con ingressi asincroni.

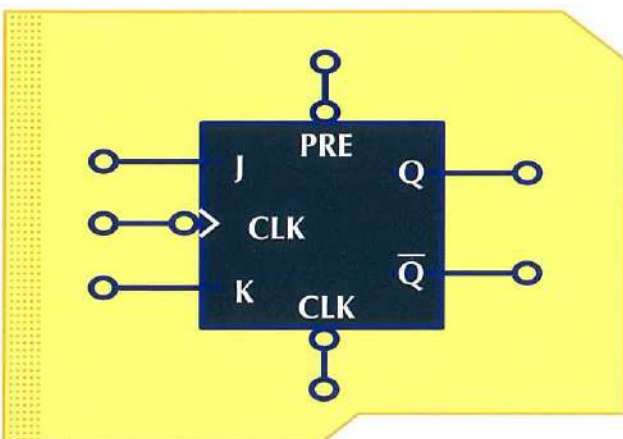
di solito si attiva a livello basso, in tal caso si utilizza un piccolo cerchio posto al termine della linea che rappresenta l'ingresso. Riassumendo, per forzare l'uscita di un bistabile a 0, si imposta a 0, in questo caso, l'ingresso CLEAR.

### Preset

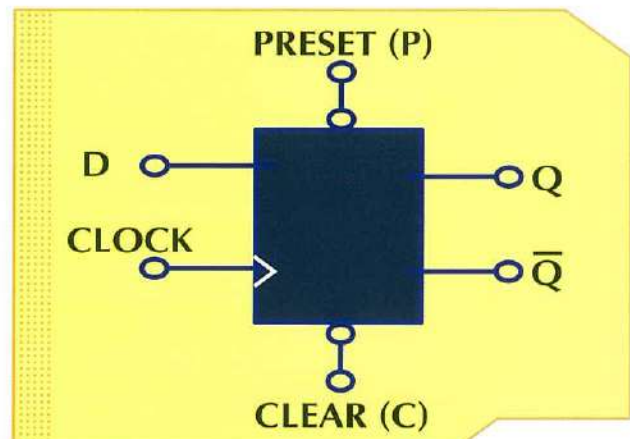
Si tratta di un ingresso asincrono che obbliga l'uscita a passare a 1 quando esso si attiva. Se è attivo a livello basso e si applica uno 0 all'ingresso PRESET, l'uscita assume il valore 1 indipendentemente dallo stato precedente e dal clock.

### Preset e Clear

Questi ingressi sono normalmente attivi a livello basso e non si devono mai attivare contemporaneamente. Se ad esempio sono attivi a livello basso, nello stato normale o di riposo devono rimanere a livello alto. Se si imposta a 0 l'ingresso PRESET, l'uscita assume immediatamente il valore 1 indipendentemente dallo stato del clock. Se invece di attivare l'ingresso PRESET attiviamo CLEAR impostandolo a 0, l'uscita passa a valore 0. Bisogna fare molta attenzione al simbolo del bistabile scelto per poter applicare su questi terminali di controllo i livelli adeguati a ogni circostanza di funzionamento.



Bistabile J-K Master Slave con ingresso asincrono T.



Bistabile tipo D con ingressi asincroni.





## Esperimenti con codice binario

**D**isponiamo già di tre circuiti stampati con componenti sufficienti per realizzare alcuni esperimenti. Sono i circuiti stampati DG01, DG02 e DG04.

Abbiamo anche un portabatterie che può alimentare l'insieme formato dai tre circuiti precedenti dopo che saranno stati assemblati tra loro.

Questa alimentazione è possibile perché è stato previsto un connettore ausiliario J44 sulla scheda del circuito DG04,

dove per il momento possiamo collegare il connettore di uscita del portabatterie.

### L'esperimento

Questo esperimento è facile da realizzare, dato che ha solamente bisogno di essere preparato. Consiste nell'applicare un codice a 4 bit all'ingresso di uno dei circuiti integrati driver 4511 che formano il circuito DG02, in modo che il risultato si visualizzi sul display delle unità.

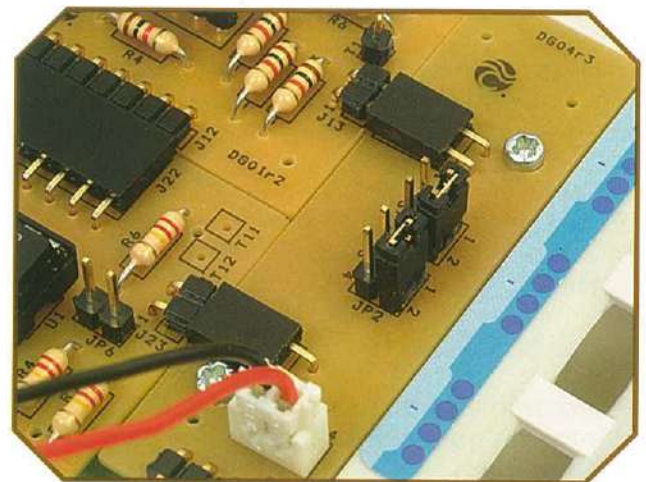
Questo esperimento può essere molto utile per fare pratica con il codice binario, dato che il display ci indicherà se abbiamo sbagliato o meno ad applicare il codice a 4 bit. Verificheremo anche l'utilizzo dei terminali ausiliari del driver LT e BI.

### Materiali

I materiali necessari sono i circuiti DG01, DG02 e DG04, con i componenti che vi abbiamo for-



Una delle prime cose da fare è verificare che le pile siano montate esattamente.



Montando i ponticelli 1 e 2 di JP1 e JP2 della scheda DG04, forniremo alimentazione ai circuiti DG01 e DG02.

nito finora già montati, in ogni caso quando le schede saranno completate l'esperimento si potrà ripetere senza alcun tipo di problema. Oltre a questi circuiti è necessario il portabatterie, tre pile da 1,5 volt tipo AA o R6 e i sei ponticelli a due terminali che vi sono stati forniti.

### Preparazione

I tre circuiti devono essere collegati tra loro, inoltre è consigliabile avviarli al laboratorio.

All'inizio della prova nessun ponticello deve essere montato né sul circuito di alimentazione DG04, né sul driver DG02, tuttavia i circuiti devono disporre, come minimo, dei componenti che possiamo vedere nelle immagini e devono essere state realizzate le saldature secondo quanto indicato nella sezione hardware.



## Alimentazione

I circuiti DG01 e DG02 si alimentano tramite il circuito DG04, benché esista la possibilità di alimentarli a 5 e a 9 volt quando il laboratorio sarà più completo, nel cui caso occorrerà montare i ponticelli JP1 e JP2 sui terminali 3 e 4 per alimentare a 9 volt, e sui terminali 1 e 2 per alimentare a 5 volt; quest'ultima posizione si utilizzerà anche per il connettore J44 del portabatterie, però in questo caso si alimenterà solamente a 4,5 volt, sufficienti per alimentare questo tipo di integrati e per realizzare l'esperimento.

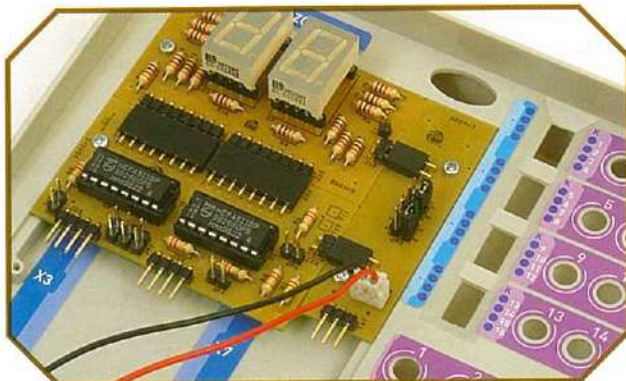
Quando il laboratorio sarà completo potremo ripetere l'esperimento a 9 V, ottenendo più luminosità sui segmenti del display.

Installando le pile bisognerà fare attenzione alla loro posizione e non ci dovremo dimenticare di montare i ponticelli che permetteranno al circuito di prendere energia dalle batterie.

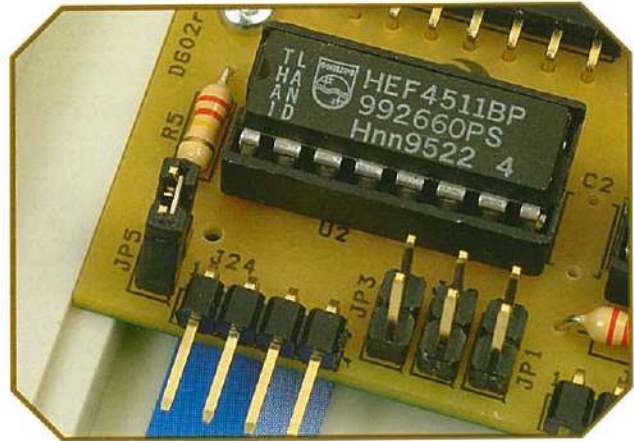
## Codice binario

Rivediamo il codice binario e la sua corrispondenza con quello decimale. In questa prova si utilizzeranno i ponticelli JP1, JP2, JP3 e JP4 che insieme alle resistenze R1, R2, R3 e R4 permetteranno di selezionare il codice binario applicato ai quattro ingressi del circuito integrato U1, un driver 4511 della famiglia CMOS, come possiamo vedere nello schema della scheda Digitale Avanzato 19.

I quattro ingressi sono simili, quando non si collega il ponticello la resistenza da 220 K stabilisce un livello logico alto all'ingresso a cui è



All'inizio della prova il circuito DG02 deve essere nella sua condizione normale, ovvero non deve avere nessun ponticello inserito.



Il ponticello per i TEST è JP5.

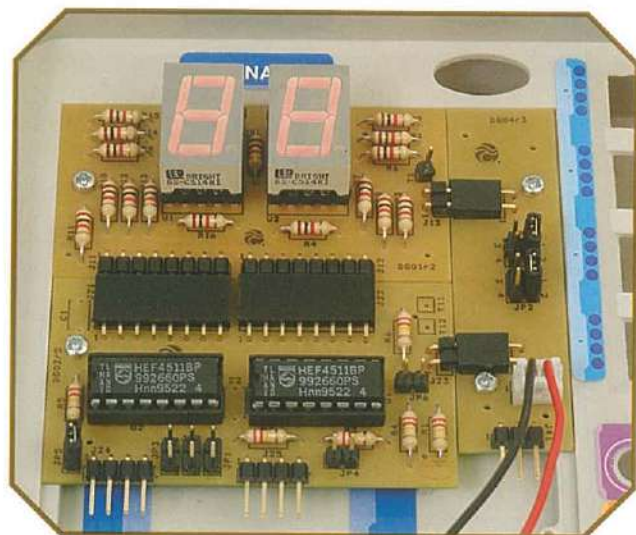
collegata e questo si interpreta come 1, inserendo però il ponticello questo ingresso passa a livello logico basso. Quando il ponticello è collegato si rappresenta con ON (0 logico), e quando non lo è con OFF (1 logico). Vediamo la tabella riassuntiva:

D C B A	JP4	JP3	JP2	JP1	Decimale	Display
0 0 0 0	ON	ON	ON	ON	0	0
0 0 0 1	ON	ON	ON	OFF	1	1
0 0 1 0	ON	ON	OFF	ON	2	2
0 0 1 1	ON	ON	OFF	OFF	3	3
0 1 0 0	ON	OFF	ON	ON	4	4
0 1 0 1	ON	OFF	ON	OFF	5	5
0 1 1 0	ON	OFF	OFF	ON	6	6
0 1 1 1	ON	OFF	OFF	OFF	7	7
1 0 0 0	OFF	ON	ON	ON	8	8
1 0 0 1	OFF	ON	ON	OFF	9	9
1 0 1 0	OFF	ON	OFF	ON	10	spento
1 0 1 1	OFF	ON	OFF	OFF	11	spento
1 1 0 0	OFF	OFF	ON	ON	12	spento
1 1 0 1	OFF	OFF	ON	OFF	13	spento
1 1 1 0	OFF	OFF	OFF	ON	14	spento
1 1 1 1	OFF	OFF	OFF	OFF	15	spento

Nota: ON ponticello inserito, OFF senza ponticello

## Test dei segmenti

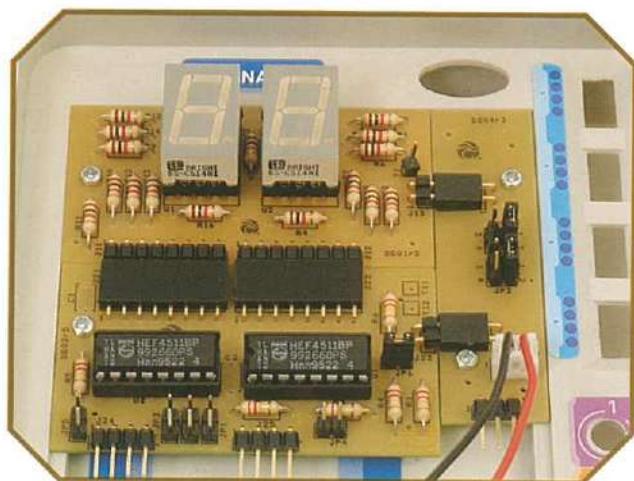
Dopo aver montato i ponticelli tra 1 e 2 di JP1 e JP2 del circuito DG04, le schede riceveranno alimentazione. Se osserviamo i display vedremo che uno dei due, quello delle unità, rimane completamente spento, mentre l'altro po-



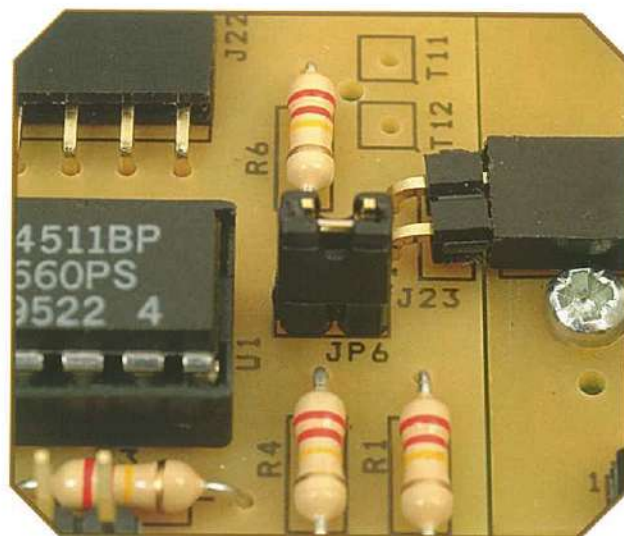
*Inserendo il ponticello su JP5 si illuminano tutti i segmenti.*

trebbe essere illuminato in modo tenue e instabile, in quanto gli ingressi del driver sono in aria e quindi non hanno un livello logico fisso.

La prima prova consiste nell'inserire il ponticello JP5 del circuito DG02, il resto dei ponticelli di questo circuito non deve essere montato. Collegando questo ponticello si devono illuminare i sette segmenti di ogni display, se qualcuno di essi non si illumina è probabile che ci sia qualche problema di collegamento, sulle saldature, sulle connessioni tra il display e lo zoccolo, oppure tra qualcuno degli integrati e il proprio zoccolo. Dopo aver realizzato questa prova si toglie il ponticello da JP5.



*Montando JP6 si impedisce la visualizzazione.*



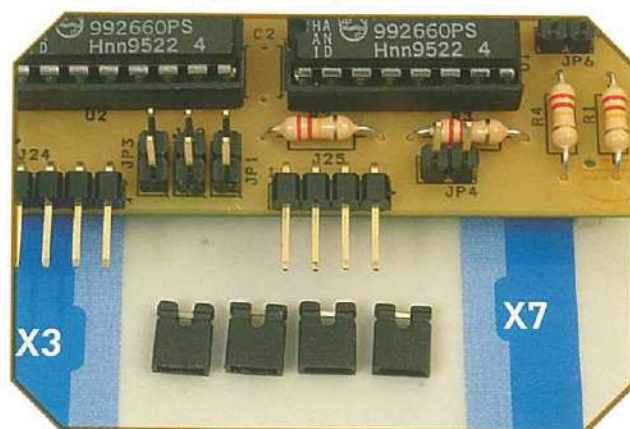
*Il ponticello per lo spegnimento totale è JP6, JP5 non deve essere inserito.*

## Spegnimento forzato

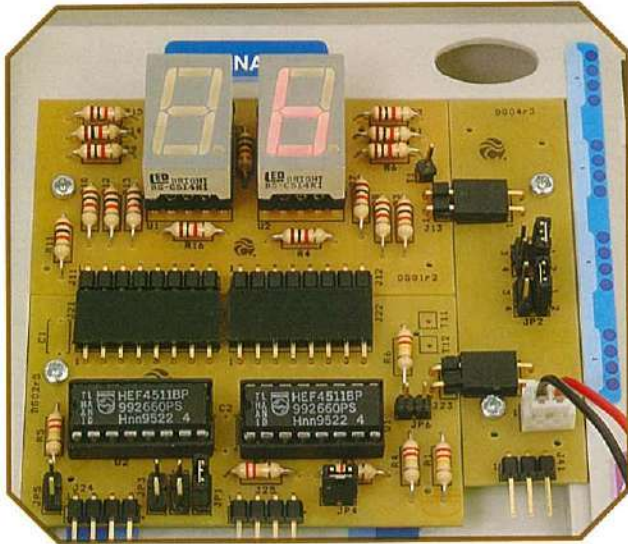
Partiamo dalla prova precedente dove abbiamo verificato che tutti i segmenti si illuminavano. Inserendo il ponticello su JP6 si devono spegnere completamente e non devono assolutamente lampeggiare. Al termine della prova bisogna estrarre il ponticello da JP6.

## Prove con codici

Benché questo circuito disponga di connettori per i codici, permette anche di inserirli in modo manuale per uno solo dei display, a questo scopo dispone di quattro ponticelli siglati



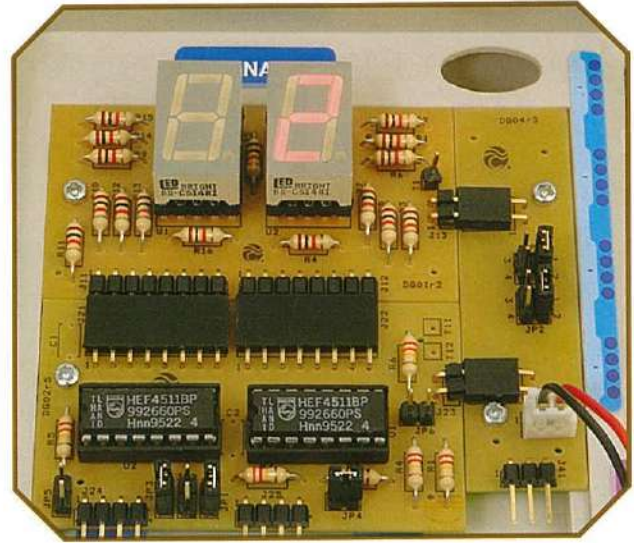
*Se si tolgono JP5 e JP6, i ponticelli da JP1 a JP4 permettono di applicare un codice al display delle unità.*



Prova con ponticelli su JP1 e JP4, si visualizza il 6.

da JP1 a JP4, dove JP1 è il bit meno significativo. Quando nessuno di questi ponticelli è installato il codice è 1111, questo valore è interpretato dal driver come errato perché corrisponde alla cifra decimale 15 e il driver accetta solamente dallo 0 al 9, quindi nessun segmento si illumina.

Iniziamo l'esperimento, per prima cosa verificiamo che i ponticelli JP5 e JP6 siano montati. Inseriamo quindi i ponticelli su JP1 e JP4, il codice quindi sarà 0110 e si deve visualizzare il 6. Ogni volta che si collega un ponticello si applica uno 0, quando non c'è il

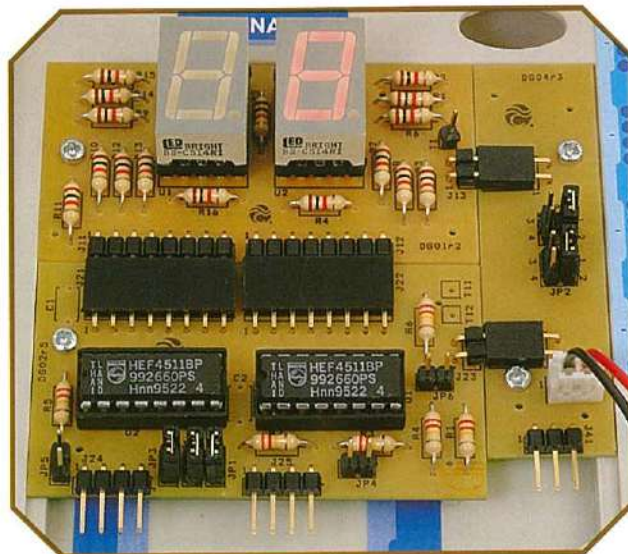


Ponticelli su JP1, JP3 e JP4 per ottenere il 2.

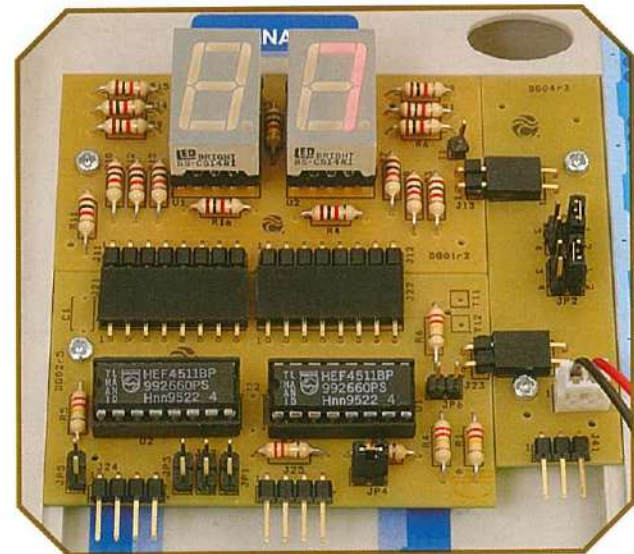
ponticello la resistenza imposta il valore 1 per default.

Possiamo ripetere fino a 10 combinazioni possibili per ottenere differenti letture, vi sono anche delle combinazioni che non sono possibili, per queste non si deve illuminare alcun segmento e sono quelle che corrispondono ai valori decimali dal 10 al 15.

Vediamo qualche esempio in più. Se montiamo i ponticelli su JP1, JP3 e JP4 si deve visualizzare il 2, e se li montiamo su JP1, JP2 e JP3 il codice sarà 1000 e verrà visualizzato il numero 8.



Ponticelli su JP1, JP2 e JP3, si visualizza il numero 8.



Ponticello su JP4, si visualizza il 7.



## Progetto con MPLAB

**P**er conoscere meglio MPLAB faremo un programma e penseremo a cosa fare con esso. Questo potente strumento permette la scrittura, la compilazione e la simulazione di un progetto. Con MPLAB voi potrete analizzare come interagisce il nostro programma con il microcontroller, come si gestisce la memoria, quali risorse utilizza, ecc. I lavori di correzione degli errori e di messa a punto risulteranno più semplici utilizzando questo strumento. In questo capitolo ci concentreremo sulla creazione del programma, lasciando la compilazione e la simulazione ai capitoli successivi.

### Definizione del progetto

Abbiamo già visto come creare un progetto in MPLAB, ma faremo comunque un rapido ripasso per poter essere sicuri di eseguire tutti i passaggi necessari. Nel menù di controllo selezioneremo l'opzione File → New, o attiveremo l'icona Crea nuovo documento sulla barra degli strumenti. Daremo poi un nome al nostro progetto e cliccheremo due volte OK, dopodiché apparirà la finestra dell'editor.

Siamo pronti per scrivere il nostro primo programma. Il linguaggio che utilizzeremo sarà l'assembler. Il nostro programma farà la somma di due valori diretti, il valore esadecimale 0x05 con 0x07, e il risultato della somma verrà caricato all'indirizzo della memoria dei dati 0x20. Tutti i file che contengono un codice in assembler avranno l'estensione ".asm".

### Concetti generali di programmazione

Sappiamo quanto sia importante una corretta metodologia di programmazione, per questo prima di iniziare a scrivere il nostro codice studieremo le regole fondamentali di programmazione in assembler.

La prima colonna dell'editor è riservata alle etichette. Le etichette sono espressioni alfanumeriche scelte dal programmatore per definire valori di indirizzi di memoria. Dovranno iniziare sempre con una lettera e non potranno essere usate espressioni riservate dell'assembler, come:

- Istruzioni.
- Direttive.
- Nomi dei registri speciali o dei bit che ne fanno parte.



Lavoriamo con MPLAB.

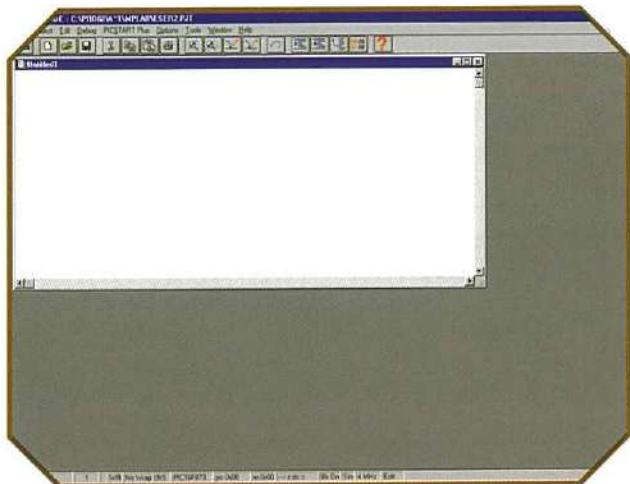
Nelle colonne successive potremo scrivere il codice mnemonico delle istruzioni o le direttive dell'assembler.

È assolutamente necessario includere commenti al programma per poterlo capire. Quando MPLAB trova un punto e virgola ";" su una linea, considera tutto ciò che viene dopo un commento. Non genererà codice macchina e non saranno considerati gli spazi bianchi o i simboli utilizzati.

È consigliabile accedere a ogni campo utilizzando il tabulatore (separeremo le colonne mediante tabulazioni), in questo modo otterremo un programma più facile da capire.

Per utilizzare maiuscole e minuscole si seguono delle regole ben precise:

- Le direttive dell'assembler si scrivono in maiuscolo.
- I nomi delle variabili si scrivono in maiuscolo.



Editor di testo dove scrivere il nostro codice.

- Gli mnemonici delle istruzioni si scrivono in minuscolo.

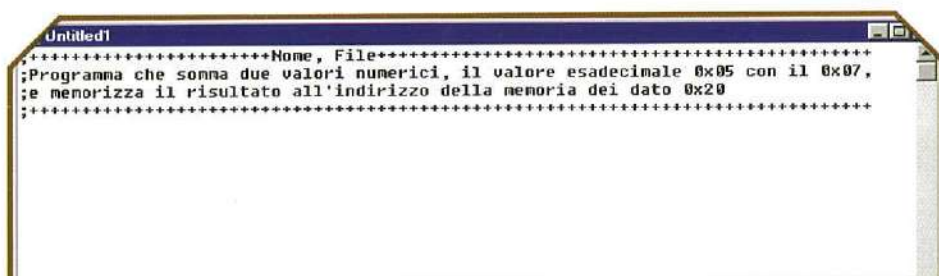
Queste regole non sono obbligatorie. Ogni programmatore ha le sue manie o le sue abitudini, però dovrà seguire una metodologia per fare in modo che i suoi programmi siano leggibili da qualsiasi altro programmatore.

## Scrittura del codice

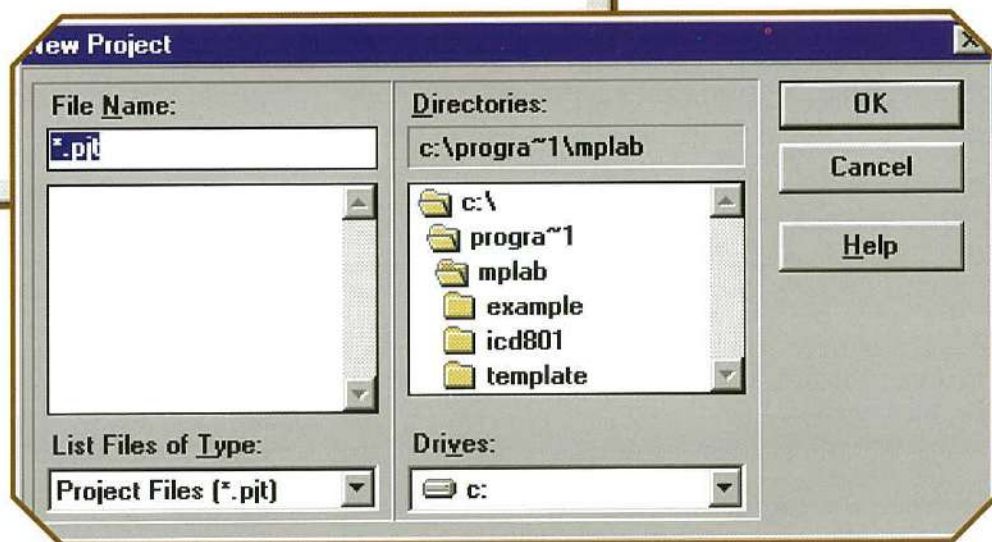
Il nostro programma inizierà con la spiegazione di ciò che si vuole fare, quindi dovremo utilizzare dei commenti. Normalmente il programmatore pone all'interno del programma il suo nome e la data in cui è stato realizzato. Tra i commenti potremo anche trovare il microcontroller utilizzato, la sua configurazione, ecc.

Mediante la direttiva LIST definiremo il microcontroller PIC che utilizzeremo nel progetto e con l'istruzione "include" includeremo nel programma un file che è una libreria dove vengono definite tutte le etichette di tutti i registri del PIC.

Queste saranno sempre le nostre due prime linee di codice in qualsiasi programma. Dovremo sempre dire al compilatore il PIC con cui vogliamo lavorare e inserire una libreria dove siano definiti i nomi di tutti i registri e i loro bit. In questo modo potremo far riferimento a essi utilizzando il nome invece dell'indirizzo che occupano nella memoria. Questo file si dovrà trovare nella directory in cui salveremo il nostro progetto, dato che il compilatore cerca in



Primi passi: indichiamo ciò che deve fare il nostro programma.



Creazione di un nuovo progetto.



```

..progra-1\mplab\ed12_1.asm
*****None, File*****
;Programma che somma due valori numerici, il valore esadecimale 0x05 con il 0x07, e memorizza
;il risultato all'indirizzo della memoria del dato 0x20
*****
        LIST  p=16F870      ;Definizione del microcontroller
        include "P16F870.inc" ;Libreria con le etichette di tutti i registri

;Dichiarazione di variabili
RISULTATO EQU 0x20        ;L'etichetta risultato si associa all'indirizzo 0x20
  
```

Stato attuale del nostro programma.

```

..progra-1\mplab\ed12_1.asm
*****None, File*****
;Programma che somma due valori numerici, il valore esadecimale 0x05 con il 0x07, e memorizza
;il risultato all'indirizzo della memoria del dato 0x20
*****
        LIST  p=16F870      ;Definizione del microcontroller
        include "P16F870.inc" ;Libreria con le etichette di tutti i registri

;Dichiarazione di variabili
RISULTATO EQU 0x20        ;L'etichetta risultato si associa all'indirizzo 0x20
ORG 0x00                 ;Vector di reset. L'istruzione successiva si carica
                        ;all'indirizzo 0x00
goto INIZIO              ;Salto all'istruzione etichettata come INIZIO

ORG 0x05                 ;Istruzione successiva su 0x05, Vector di Interrupt

INIZIO: movlw 0x05         ;Muove sul registro W il valore numerico 0x05
      addlw 0x07         ;Somma al contenuto di W il valore 0x07, depositando
                        ;il risultato su W
      movwf RISULTATO   ;Sposta il contenuto di W sulla variabile RISULTATO]
  
```

Il programma praticamente terminato.

questa directory tutti i file che sono stati inclusi nel programma da compilare.

Potete aprire il file "P16F870.inc" che si trova sul CD e osservare come vengono definiti i registri.

Dopo queste due linee di codice abbiamo la zona delle dichiarazioni delle variabili. Prima di iniziare il programma, dobbiamo dichiarare tutte le variabili che utilizzeremo all'interno dello stesso. Per questo dobbiamo nominarle e assegnare loro una zona di memoria. Nel nostro progetto dobbiamo definire la variabile risultato, dove scriveremo il risultato dell'operazione. Poiché l'indirizzo della memoria ci è stato imposto dalle specifiche di progetto, non ci resta che mettere in relazione il nome con l'indirizzo e questo si fa con la direttiva EQU.

Nella figura possiamo osservare la forma che sta acquisendo il nostro programma.

Ciò che viene di seguito verrà utilizzato anche in tutti i programmi che faremo. Bisogna definire dove si troverà il codice da eseguire e in quale zona della memoria di codice si trova la routine di interrupt, nel caso sia presente.

Quindi, mediante la direttiva ORG, definiamo la posizione iniziale del codice del nostro programma. Con l'istruzione di salto goto indichiamo dove deve essere la destinazione del salto incondizionato.

A partire dall'etichetta di INIZIO potremo scrivere il codice riferito alle specifiche

```

*****None, File*****
;Programma che somma due valori numerici, il valore esadecimale 0x05 con il 0x07, e memorizza
;il risultato all'indirizzo della memoria del dato 0x20
*****
        LIST  p=16F870      ;Definizione del microcontroller
        include "P16F870.inc" ;Libreria con le etichette di tutti i registri

;Dichiarazione di variabili
RISULTATO EQU 0x20        ;L'etichetta risultato si associa all'indirizzo 0x20
ORG 0x00                 ;Vector di reset. L'istruzione successiva si carica
                        ;all'indirizzo 0x00
goto INIZIO              ;Salto all'istruzione etichettata come INIZIO

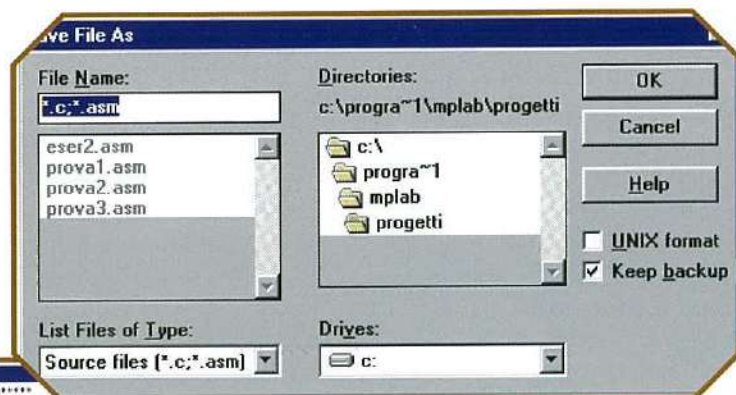
ORG 0x05                 ;Istruzione successiva su 0x05, Vector di Interrupt

INIZIO: movlw 0x05         ;Muove sul registro W il valore numerico 0x05
      addlw 0x07         ;Somma al contenuto di W il valore 0x07, depositando
                        ;il risultato su W
      movwf RISULTATO   ;Sposta il contenuto di W sulla variabile RISULTATO]
  
```

Libreria "P16F870.inc" dove si trovano le definizioni di tutti i registri.



Salviamo il codice.



```

c:\progra~1\mplab\progetti\2_1.asm
.....None, file
;Programma che somma due valori numerici, il valore esadecimale 0x05 con il 0x07, e memorizza
;il risultato all'indirizzo della memoria del dato 0x20
;.....
LIST      p=16F870      ;Definizione del microcontroller
include  "16F870.inc"  ;Libreria con le etichette di tutti i registri

;Dichiarazione di variabili
RISULTATO EQU 0x20      ;L'etichetta risultato si associa all'indirizzo 0x20
ORG 0x00      ;Vector di reset, l'istruzione successiva si carica
;all'indirizzo 0x00
goto INIZIO   ;Salto all'istruzione etichettata come INIZIO
ORG 0x05      ;Istruzione successiva su 0x05, Vector di Interrupt
INIZIO:      movlw 0x05      ;Muove sul registro W il valore numerico 0x05
;Somma al contenuto di W il valore 0x07, depositando
;il risultato su W
addlw 0x07
movwf RISULTATO ;Sposta il contenuto di W sulla variabile RISULTATO

STOP        nop          ;Non fa niente però riserva queste posizioni per
;collocare un punto di arresto
nop
END         ;Direttiva che indica la fine del programma sorgente

```

Codice finale risultante.

del progetto. Anche se non conosciamo il repertorio delle istruzioni del nostro microcontroller, possiamo risolvere il programma con tre sole istruzioni. Mediante l'istruzione `movlw` carichiamo un valore sul registro di lavoro `W`, con l'istruzione `addlw` sommiamo il valore indicato nell'operando al registro di lavoro. Il risultato dell'operazione viene scritto nel registro `W`, sovrascrivendo il valore precedente. Dobbiamo quindi spostare il risultato alla variabile che abbiamo definito e, a questo scopo, utilizzeremo l'istruzione `movwf`.

Nella figura possiamo vedere come diventa il programma. Non ci resta che terminarlo, ovvero trasmettere al microcontroller l'ordine di terminare l'esecuzione del codice. I programmi si terminano con la direttiva `END`. Per una successiva analisi del programma inseriremo delle istruzioni che non fanno nulla, `nop` (No Operation), in questo modo potremo simulare il programma e verificare che funzioni realmente come desideriamo. Potremmo anche utilizzare un'istruzione di `sleep` in modo che il microcontroller, o il simulatore nel nostro caso, si fermi in stato di riposo o addormentato.

Il programma risultante si può vedere nella figura.

## Che cosa facciamo con il programma?

Dopo aver terminato il programma, per prima cosa lo dobbiamo salvare. Conosciamo già questo processo, ma vale la pena di ripassarlo. Selezioneremo `File` e `Save` e apparirà la finestra di dialogo della figura, dove daremo un nome al file creato. Possiamo chiudere il programma, salvando i cambiamenti fatti nel progetto precedente, dato che per il momento non lavoreremo più con esso.

Nei capitoli successivi faremo pratica con questo programma con le opzioni di compilazione e simulazione. Compilando si convertirà il codice creato in assembler in codice macchina e, nel frattempo, scopriremo l'eventuale presenza di errori nel nostro codice. Simulando potremo verificare che il programma creato risponda alle esigenze iniziali del progetto. Nei fascicoli successivi lavoreremo con queste opzioni e potremo verificare quanto sia semplice gestire questo software e quanto risulti utile.

L'obiettivo è imparare a utilizzare questo potente strumento, dato che risulta fondamentale la programmazione dei microcontroller PIC.





# Compilazione con MPLAB

**C**i sono errori nel nostro programma? Funzionerà come vogliamo? Cosa succede se cambio questa istruzione con un'altra? Tutte queste domande possono trovare una risposta su MPLAB. Finora abbiamo editato un programma, abbiamo visto la configurazione e una piccola introduzione, dobbiamo però addentrarci in questo software e iniziare a gestire alcune delle sue funzioni più importanti.

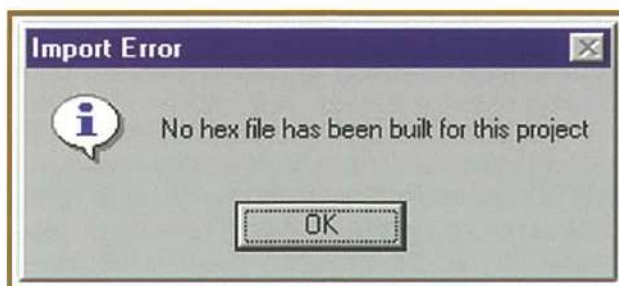
## Prima di compilare

Per lavorare utilizzeremo l'esercizio del capitolo precedente di MPLAB. Dobbiamo aprire il nostro progetto selezionando sul menù di controllo Project → Open Project.

Sceglieremo l'ultimo progetto che abbiamo fatto all'interno della cartella di lavoro e cliccheremo OK. Ci apparirà un messaggio di errore il quale ci indica che non esiste alcun file in codice macchina associato al progetto.

Accetteremo l'errore e apparirà la videata con l'editor e il codice che abbiamo preparato la volta precedente. Il passo successivo sarà quello di tornare a editare il nostro progetto selezionando sul menù di controllo Project → Edit Project, con la conseguente comparsa della finestra che possiamo vedere nella figura della pagina successiva. Se clicchiamo sul file che appare con l'estensione tra parentesi quadre, in questo caso è ese2[.hex], si attiverà la possibilità di selezionare Node Properties (prima era disabilitato).

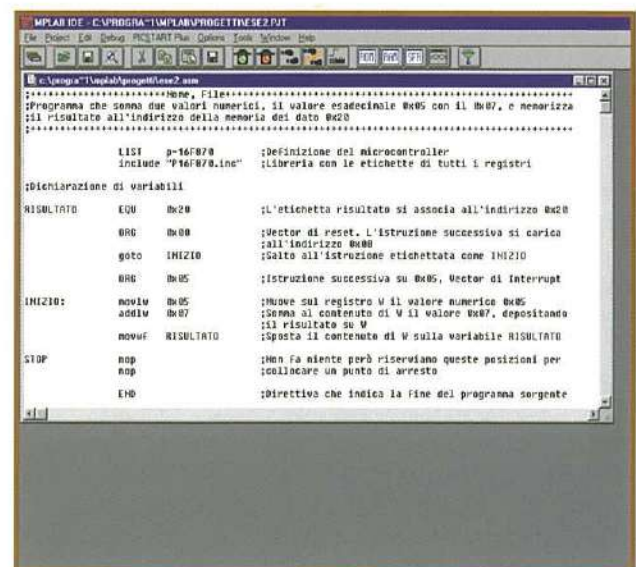
Scegliendo questa opzione accederemo a una finestra da dove si selezioneranno i file e i formati che si otterranno assemblando il



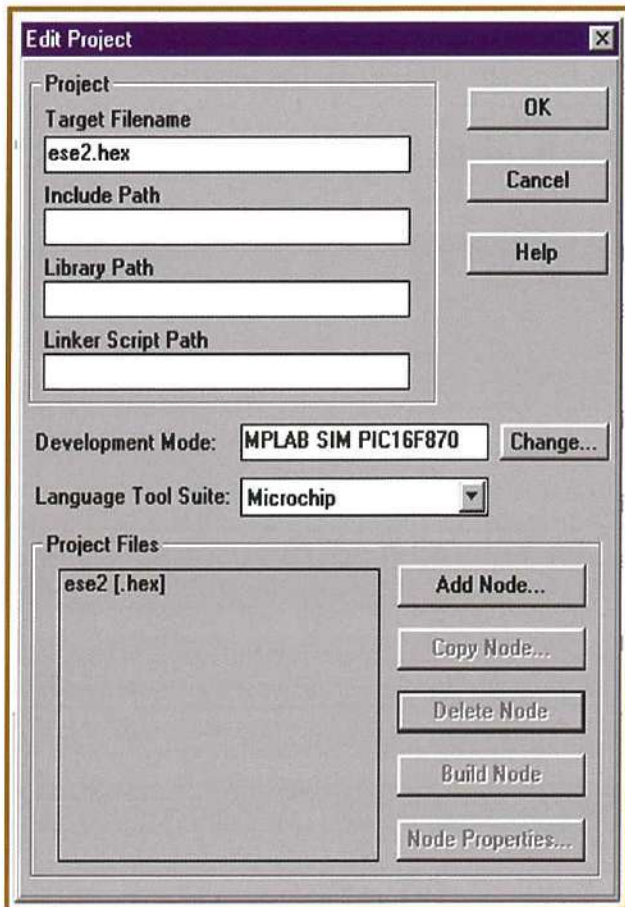
Messaggio del programma all'apertura del progetto.

programma. Non occorre cambiare nulla, dobbiamo solo sapere che abbiamo selezionato la creazione di un file di errore (error file) durante la procedura di assemblamento, un file con il listato del programma (list file) e abbiamo impostato la sensibilità all'utilizzo delle maiuscole (case sensitivity). Cliccheremo OK per continuare e torneremo alla videata precedente in cui selezioneremo Add Node.

Nella finestra che appare sceglieremo il file .asm contenente il codice necessario per sommare i due numeri che erano stati proposti come esercizio nel capitolo precedente. In seguito potremo verificare che nel campo Project Files è stato aggiunto il file selezionato. Cliccheremo OK e torneremo alla videata con il programma scritto. Siamo pronti per la compilazione.



Partiamo dall'esercizio di somma di due numeri.



Videata di edizione del progetto.

## Strumenti di compilazione

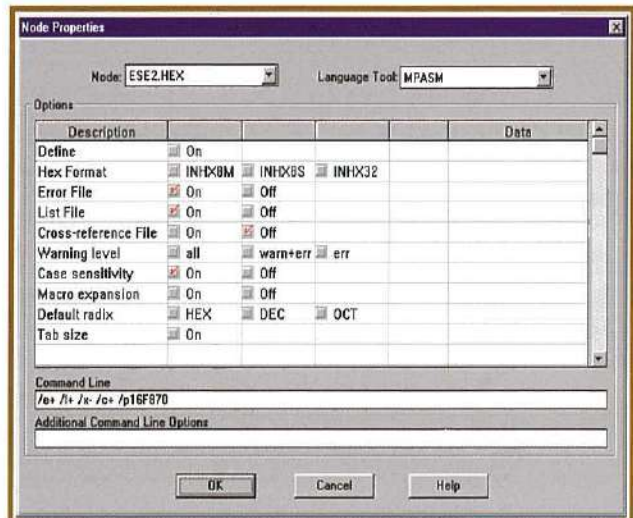
Possiamo commutare tra diverse barre di strumenti in MPLAB, una delle quali è specifica per il lavoro con i progetti e la compilazione. Nella figura della pagina successiva possiamo vedere la barra degli strumenti di compilazione a cui facevamo riferimento con tutte le sue funzioni in dettaglio. Vi consigliamo di realizzare nuovamente tutti i passaggi visti nel capitolo precedente, utilizzando però i pulsanti della barra degli strumenti.

## La compilazione

Se non abbiamo eseguito i passaggi precedenti, in cui abbiamo aggiunto il file in assembler al progetto cercando di assemblare, apparirà il messaggio di errore della figura. MPLAB ci indica mediante questo messaggio che non è stato specificato alcun file come sorgente.



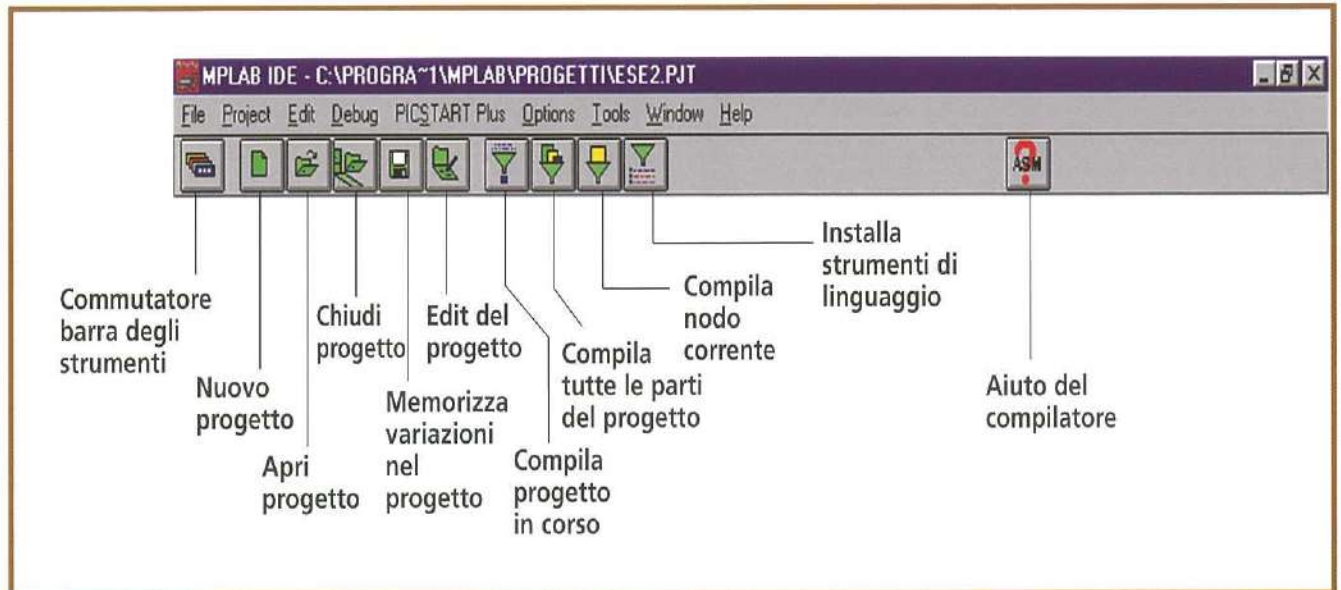
Messaggio di errore causato dalla mancanza del file assembler all'interno del progetto.



Proprietà del nodo per l'assemblaggio.

Noi abbiamo seguito tutti i passaggi, quindi potremo compilare direttamente. Per assemblare il programma dobbiamo selezionare nel menù di controllo l'opzione Project → Build All o, utilizzando la barra degli strumenti, cliccare il pulsante corrispondente a Compilazione congiunta del progetto.

Quando un programma non ha errori, il risultato della compilazione sarà quello indicato nella figura. Apparirà una videata in cui ci verrà comunicato che la compilazione è stata eseguita con successo. A partire da questo momento, nella directory di lavoro dove si trovavano il progetto e il file in assembler, si troveranno anche il file con il listato del codice, il file degli errori e il file con il codice macchina (estensione .hex). Quest'ultimo file sarà quello da scrivere sul PIC e farà girare su di esso l'applicazione che abbiamo creato. Utilizzeremo sempre Build All, dato che questa opzione include la compilazione di tutta la struttura



Barra degli strumenti di progetti e compilazione.

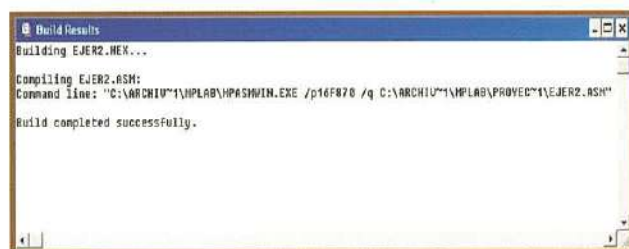
del progetto. Il resto delle opzioni Make Project e Build Node non contemplano l'intero progetto.

### Gli errori

Quando programiamo, una delle domande che ci poniamo è: cosa succede quando in un programma ci sono degli errori?

Se in un programma ci sono errori MPLAB li rileva e impedisce la compilazione. Inseriremo alcuni errori nel nostro programma. Nella figura possiamo osservare come siano stati introdotti tre errori: l'omissione del punto e virgola su una linea che volevamo fosse un commento, l'inserimento di un simbolo nell'operando che indica un indirizzo e la presenza di un carattere nello mnemonico di un'istruzione.

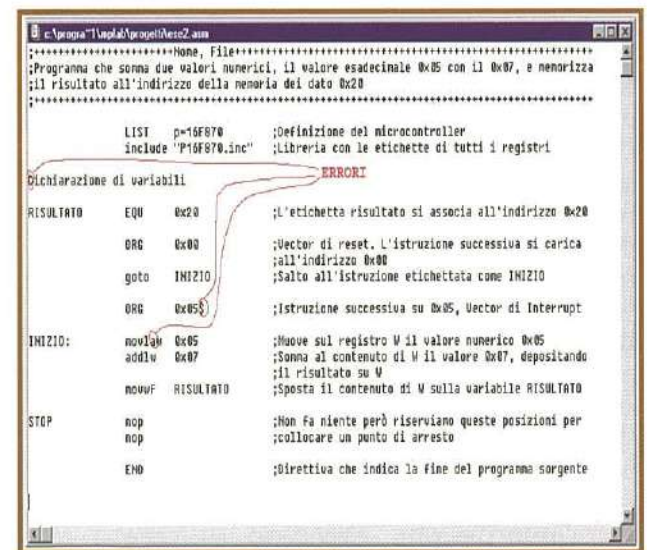
Quando si realizza una modifica all'interno del codice, si devono salvare i cambiamenti prima di eseguire la compilazione. Quindi



Videata del risultato di una compilazione senza errori.

selezioneremo Project → Save Project o il pulsante della barra degli strumenti corrispondente, preparandoci per compilare. Quando lanceremo la compilazione, il risultato di questa azione è una videata come quella mostrata nella figura della pagina successiva. Questa videata indica la presenza di errori e comunica che la compilazione non ha avuto luogo.

Se ci posizioniamo con il mouse sulla prima linea degli errori e clicchiamo due volte, il cur-



Errori inseriti nel codice.



```
Build Results
Building ESE2.HEX...

Compiling ESE2.ASH:
Command Line: "C:\PROGRAMMI\MPLAB\MPASMIN.EXE /e+ /i+ /x- /c+ /p16F870 /q C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH"
Error[122] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 9 : Illegal opcode (dl)
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 15 : Symbol not previously defined (INIZIO)
Error[112] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 17 : Missing operator
Error[122] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 19 : Illegal opcode (movlw)
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 20 : Overwriting previous address contents (0000)
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 20 : Overwriting previous address contents (0000)

MPLAB is unable to find output file "ESE2.HEX". This may be due to a compile, assemble, or link process failure.
Build Failed.
```

*Risultato finale del tentativo di compilazione.*

sore si posizionerà sulla linea di codice che contiene l'errore. Il messaggio che appare nella finestra spiega in che cosa consiste l'errore, anche se è in inglese, specifica cos'è che non va. In questo primo errore informa che è stato trovato un simbolo non definito in precedenza, una parola sconosciuta. Posizionandoci con il cursore su questa linea e con l'aiuto del messaggio che ci fornisce il compilatore, possiamo identificare chiaramente l'errore e correggerlo. Torneremo quindi a scrivere il punto e virgola all'inizio, salveremo il programma e proveremo nuovamente a compilare. Questa volta vedremo che gli errori si sono ridotti, ora il compilatore ci informa della presenza di cinque errori.

Se ripetiamo la stessa operazione con il primo errore inserito, vedremo che prima dell'etichetta di INIZIO c'è un simbolo strano. Questo non permette il riconoscimento dell'etichetta. Per risolvere togliamo questo simbolo, salviamo nuovamente e torniamo a compilare. Gli errori si sono ridotti a due. A volte il compilatore, in presenza di un errore, ci informa di tutto ciò che trova di estraneo o non valido per la compilazione. Per questo, un semplice errore come quello corretto in precedenza può provocare tre rilevamenti di errore da parte del compilatore o tre ragioni che impediscono al programma di essere compilato correttamente.

Osserviamo i due errori rimanenti, vedremo chiaramente che abbiamo utilizzato uno mnemonico sbagliato. Correggendo l'errore

```
Build Results
Building ESE2.HEX...

Compiling ESE2.ASH:
Command Line: "C:\PROGRAMMI\MPLAB\MPASMIN.EXE /e+ /i+ /x- /c+ /p16F870 /q C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH"
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 15 : Symbol not previously defined (INIZIO)
Error[112] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 17 : Missing operator
Error[122] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 19 : Illegal opcode (movlw)
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 20 : Overwriting previous address contents (0000)
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 20 : Overwriting previous address contents (0000)

MPLAB is unable to find output file "ESE2.HEX". This may be due to a compile, assemble, or link process failure.
Build Failed.
```

*Dopo aver corretto il primo errore rimangono da correggere gli altri.*

torneremo alla situazione iniziale in cui il programma si compila correttamente e, quindi, genera i file necessari per caricare il programma sul microcontroller.

## Conclusioni

Applicare questi nuovi concetti servirà come esercizio. Sarà possibile creare un progetto con il codice dell'esercizio precedente e cercare di compilare senza inserire il file assembler all'interno del progetto. Se inserirete degli errori nel programma potrete familiarizzare con il modo di cui dispone MPLAB di indicare gli errori; in definitiva, è consigliabile provare tutto ciò che serve per affinare le nuove conoscenze acquisite.

```
Build Results
Building ESE2.HEX...

Compiling ESE2.ASH:
Command Line: "C:\PROGRAMMI\MPLAB\MPASMIN.EXE /e+ /i+ /x- /c+ /p16F870 /q C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH"
Error[110] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 15 : Symbol not previously defined (INIZIO)
Error[122] C:\PROGRAMMI\MPLAB\PROGETTI\ESE2.ASH 19 : Illegal opcode (movlw)

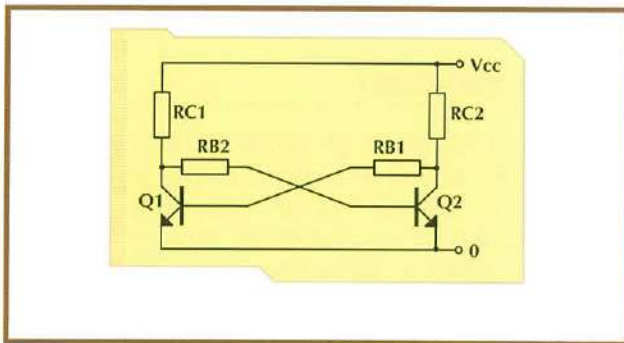
MPLAB is unable to find output file "ESE2.HEX". This may be due to a compile, assemble, or link process failure.
Build Failed.
```

*Dopo aver corretto il secondo errore il compilatore rileva solamente due errori.*

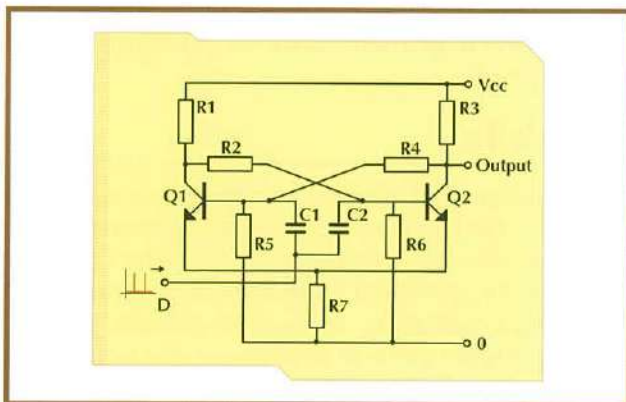


# Multivibratori con transistor

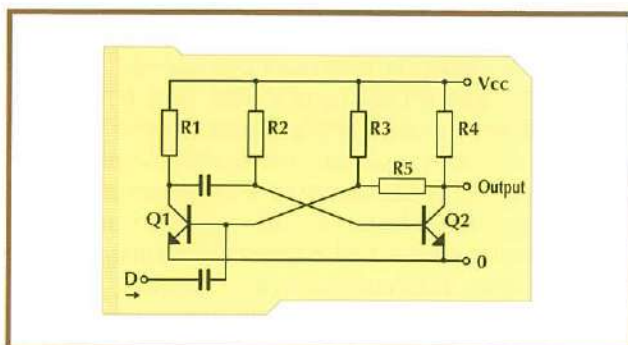
**I** multivibratori a transistor sono circuiti di commutazione in cui l'uscita è in parte determinata dalle variabili d'ingresso e in parte dalle uscite stesse. Il segnale di uscita, o parte di esso, viene riportato sull'ingresso, in altre parole si esegue una retroazione sull'ingresso, quindi si tratta di circuiti retroazionati. Osservando gli schemi è possibile apprezzare questo nel dettaglio. L'uscita del circuito dipende, quindi, dallo stato precedente dell'uscita stessa e dall'eccitazione che si applica all'ingresso.



Circuito bistabile di base.



Circuito bistabile con transistor.



Circuito monostabile con transistor.

## Il transistor

In questi circuiti il transistor lavora in commutazione, ovvero, o si trova in interdizione oppure in saturazione, quindi conduce o non conduce e questo ci permette di studiare i circuiti in un modo abbastanza semplice.

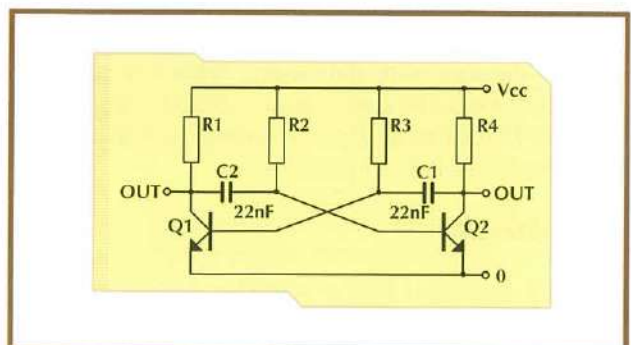
## Tipi di circuiti

Questi circuiti possono rimanere in uno o due stati stabili, a seconda degli ingressi e delle uscite, e del loro stato precedente. Vengono anche definiti degli stati instabili in cui il circuito può rimanere per un periodo di tempo e, grazie ai componenti interni, al termine di questo periodo di tempo ritorna in uno stato stabile.

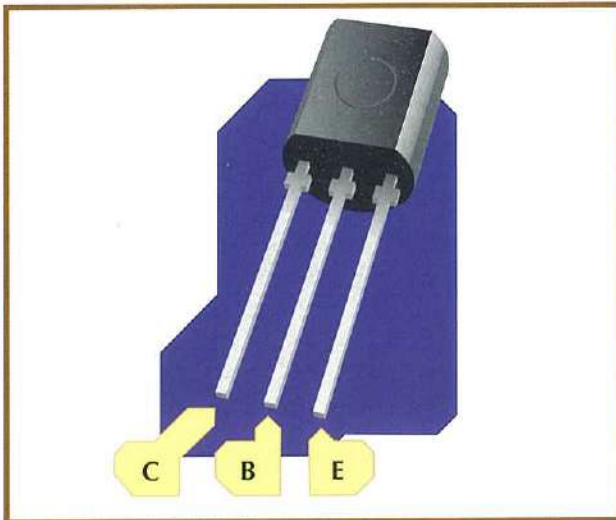
I circuiti multivibratori si possono classificare in tre tipi: bistabili, monostabili e astabili.

## Bistabile

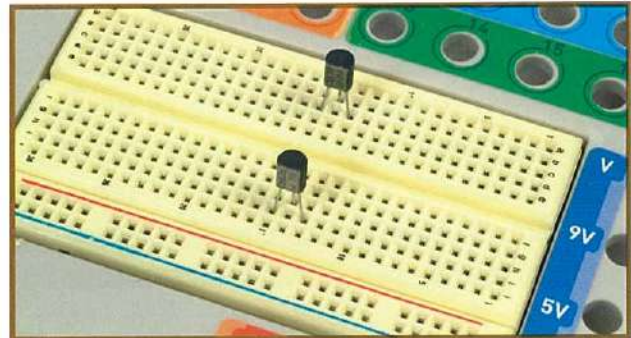
Questo circuito ha due stati stabili e, nella condizione di riposo, può rimanere in qualsiasi di questi fino a quando non riceve un segnale esterno che lo faccia cambiare di stato.



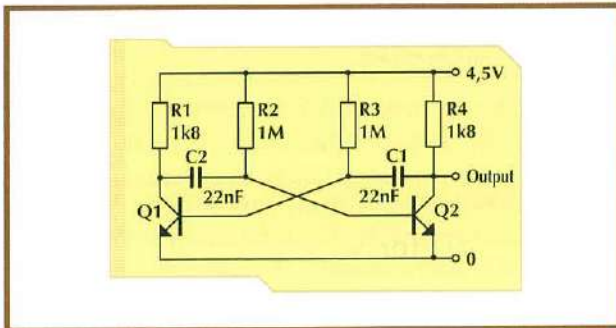
Circuito astabile con transistor.



Terminali del transistor BC547 o BC548.



Transistor inseriti.



Esempio di un circuito astabile con transistor.

## Monostabile

Questo multivibratore ha uno stato stabile e un altro instabile. Lo stato instabile si ottiene grazie a un componente che può immagazzinare energia per un certo tempo, normalmente un condensatore. Quando si attiva lo stato instabile il sistema rimane in questo stato per un periodo di tempo che dipende dai componenti e dal circuito utilizzato; trascorso questo tempo, il circuito passa a uno stato stabile e vi rimane fino a quando non verrà nuovamente forzato sullo stato instabile.

## Astabile

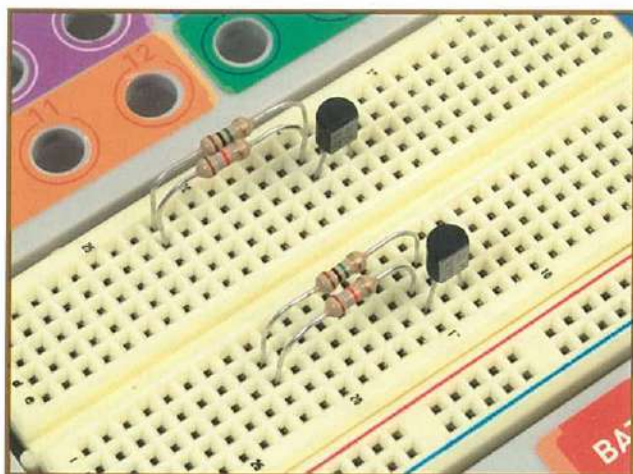
Non ha nessuno stato stabile, però possiede due stati instabili entro i quali cambia continuamente, quindi sulla sua uscita otterremo un segnale periodico.

## Multivibratori

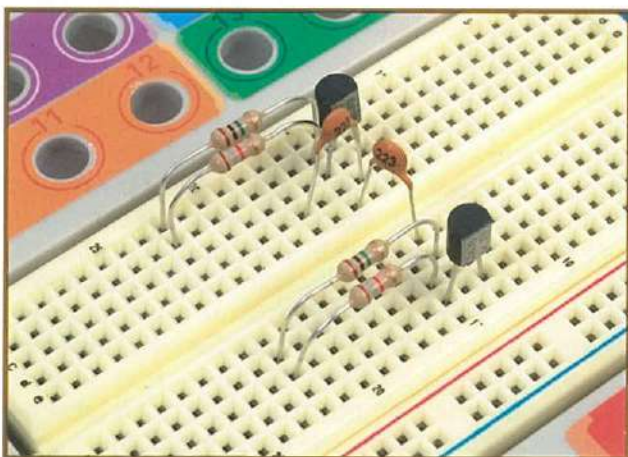
Fondamentalmente gli schemi dei tre circuiti sono molto simili, almeno a prima vista. Contengono due amplificatori invertenti con accoppiamento RC, ognuno dei quali è formato da un transistor e dalle rispettive resistenze di polarizzazione. Vi ricordiamo che quando un transistor è in saturazione conduce, e la sua tensione di collettore-emettitore è molto bassa, quasi nulla, mentre quando è in interdizione non conduce e la sua tensione collettore-emettitore è praticamente la tensione di alimentazione.

## Bistabile

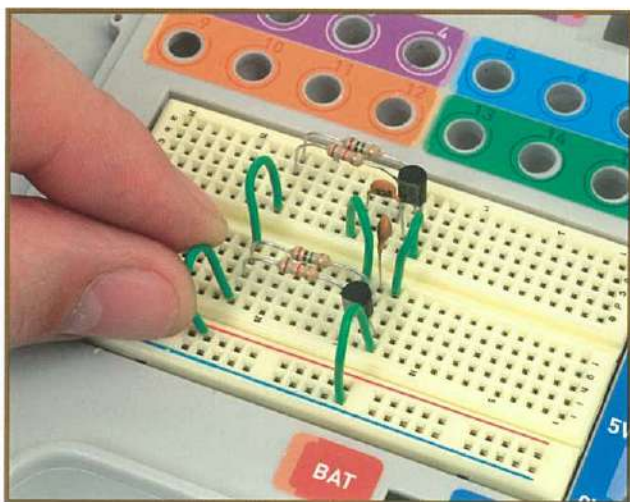
Se osserviamo lo schema vedremo i due transistor Q1 e Q2 e le loro resistenze di polarizzazione, oltre alle due resistenze R2 e R4 che collegano le uscite, ovvero il collettore di un transistor sulla base dell'altro, o, in altre parole, sull'ingresso; queste sono le resistenze che generano la retroazione. Quando si collega l'alimentazione al circuito, i transistor Q1 e Q2 si polarizzano nel seguente modo: le resistenze R1, R2 e R6 polarizzano la base del transistor Q2, mentre le resistenze R3, R4 e R5 fanno lo stesso con Q1. Come sempre esistono piccole differenze da un circuito all'altro, uno dei due transistor conduce prima dell'altro, supponiamo che sia Q1, che entra in saturazione, quindi la tensione sul suo collettore si abbassa e diminuisce anche sulla base di Q2, impedendo a questo di entrare in conduzione e obbligandolo a rimanere in interdizione. Il circuito rimane in questo stato fino a quando non si produce l'attivazione e si inverte la situazione, scambiando lo stato tra entrambi i transistor.



Transistor e resistenze montate.



Inserzione dei condensatori.



Installazione dei fili di connessione.

## Monostabile

Osservando il circuito vedremo che è simile al precedente, nella condizione di riposo, ovvero quando l'alimentazione rimane collegata per un determinato tempo e il circuito è in uno stato stabile, il transistor Q1 è in interdizione e Q2 è saturo. Quando, però, si applica un impulso di attivazione alla base del transistor Q1, questi entra in conduzione, si satura e la sua tensione di collettore diventa molto bassa, scaricando rapidamente il condensatore C1 e portando Q2 in interdizione.

Al termine dell'impulso di attivazione inizia nuovamente la carica del condensatore C1 in modo lento, tramite la resistenza R2, fino a quando la tensione sulla base di Q2 lo farà condurre ed entrare in saturazione, obbligando il transistor Q1 a passare in interdizione e tornando quindi nello stato stabile fino al prossimo impulso di attivazione.

## Astabile

Guardando lo schema possiamo vedere che il circuito è totalmente simmetrico. Anche se la tensione si applica simultaneamente su entrambi i transistor, vi sono piccole differenze che permetteranno a uno dei due di saturarsi obbligando l'altro a entrare in interdizione. Quando uno dei transistor è in interdizione, supponiamo Q1, la sua tensione di collettore sale e il condensatore C1 si carica fino a portare Q1 in conduzione; quando questo si satura, la carica del condensatore C2 scende di colpo facendo aprire il transistor Q2. Dato che il circuito è simmetrico, il processo si ripete nuovamente e si ottiene così un segnale periodico.

## Esperimento

Dato che al momento disponiamo già degli elementi sufficienti, proviamo a costruire un astabile con i transistor. A questo scopo abbiamo bisogno dei seguenti componenti:

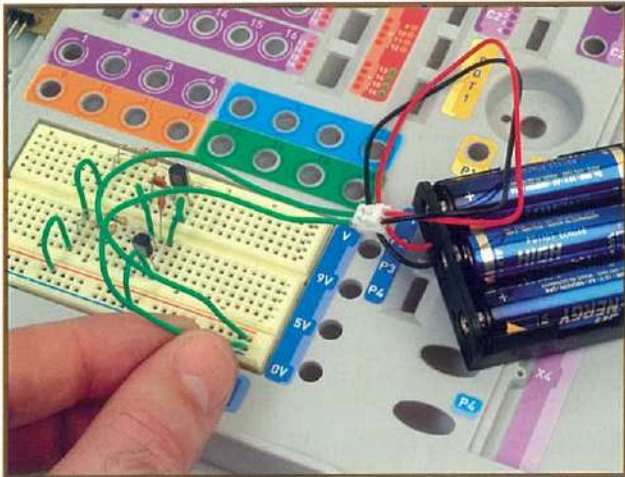
### ASTABILE CON TRANSITOR

Q1, Q2 Transistor NPN BC547 o BC548

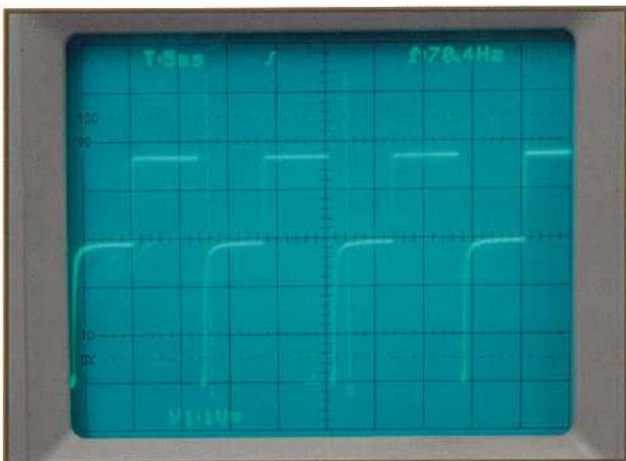
R1, R4 Resistenza 1K8 (marrone, grigio, rosso)

R2, R3 Resistenza 1 M (marrone, nero, verde)

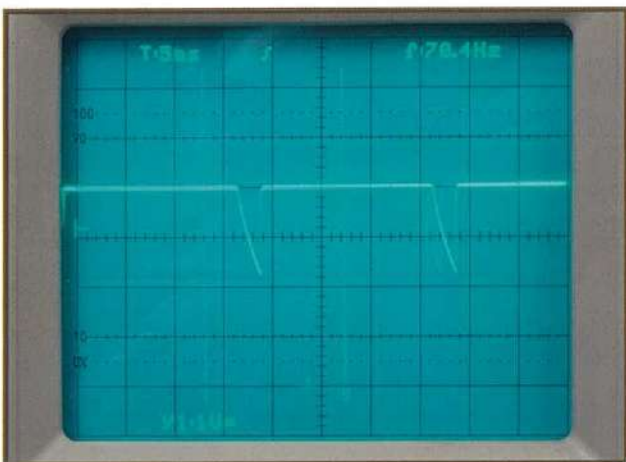
C1, C2 Condensatore 22 nF



Collegamenti provvisori dell'alimentazione.



Segnale di uscita sull'oscilloscopio.



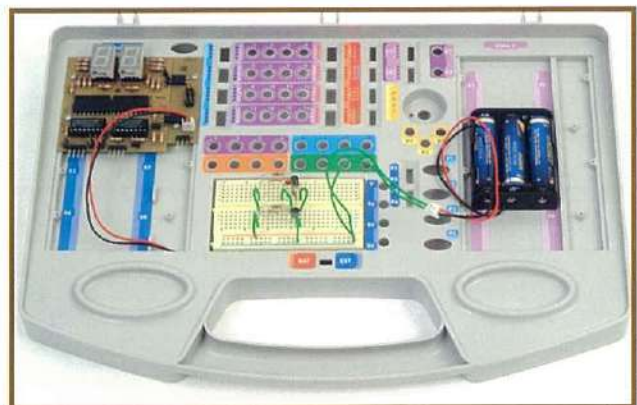
Tensione ai capi del condensatore C1.

## Astabile con transistor

Utilizzeremo il circuito di esempio e monteremo i componenti indicati sulla scheda Bread Board. Dato che si tratta del primo esperimento lo descriveremo in modo dettagliato, passo a passo, aiutandoci con le fotografie e facendo molta attenzione al montaggio di ogni componente. Per prima cosa inseriremo i due transistor osservando attentamente il loro contenitore per evitare errori. Poi posizioneremo le resistenze, che si identificano grazie alle bande colorate del loro contenitore. Non potremo scambiare le une con le altre perché hanno valori diversi, bisognerà inserire i loro terminali in modo verticale per assicurare il contatto.

A questo punto posizioneremo i condensatori. I collegamenti che non si possono realizzare utilizzando la scheda Bread Board verranno fatti con pezzi di filo.

In questo caso l'alimentazione sarà collegata con altri due pezzi di filo al portabatterie che non è ancora stato montato, il cavo rosso è il positivo e il nero è il negativo, bisogna fare molta attenzione per non sbagliare. Per non commettere errori, sulla scheda Bread Board si utilizzerà il rosso per il positivo e l'azzurro per il negativo, nel nostro schema gli emettitori dei transistor devono essere collegati al negativo dell'alimentazione. Questo circuito, tuttavia, non si può provare, perché bisogna attendere che il laboratorio sia più completo. Nel prossimo numero e in quello successivo, realizzeremo un esperimento simile che potremo provare, ma per il momento, l'unico modo per vedere le forme d'onda è collegare un oscilloscopio che ci mostrerà il risultato ottenuto.



Aspetto dell'esperimento.





## Astabili con il 4093

**In questo esperimento o, per meglio dire, gruppo di esperimenti, si realizzano generatori a onda quadra con il circuito integrato CMOS 4093. Il funzionamento del circuito si può verificare facendo in modo che la frequenza di questi generatori sia tale da poter essere percepita dall'occhio umano, in modo da poter utilizzare un diodo LED come indicatore di funzionamento.**

### Il 4093

La prima parte dell'esperimento consiste nel verificare il funzionamento del circuito integrato. Basta ricordare la tabella della verità di una porta NAND: per fare in modo che l'uscita sia zero, entrambi gli ingressi devono essere a livello 1, per le altre tre possibili combinazioni l'uscita è 1.

### Montaggio

Per questa prima prova è sufficiente inserire il circuito integrato nella zona centrale della scheda Bread Board e collegare un LED su una delle sue uscite, ad esempio sul terminale 10, inserendo una resistenza da 1K8.

Il catodo del LED deve rimanere sempre verso il positivo dell'alimentazione, la resistenza può essere collegata tra il terminale dell'integrato e l'anodo del LED oppure tra il catodo del LED e il negativo dell'alimentazione. Per quanto riguarda gli ingressi, per ottenere un 1 si collega con un filo al positi-

vo dell'alimentazione, e per ottenere uno 0 si collega al negativo.

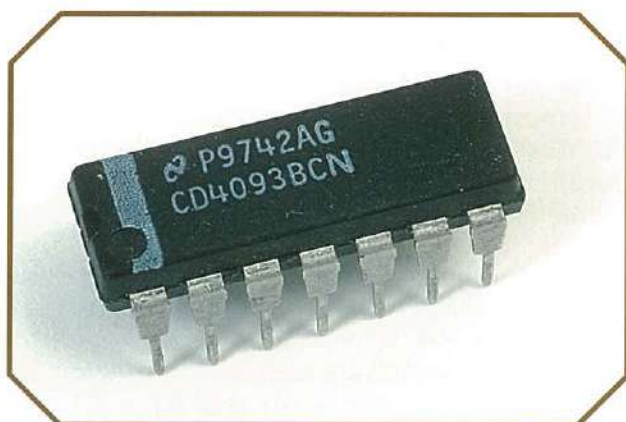
### Alimentazione

Questo tipo di circuiti CMOS può essere alimentato tra 3,5 e 12 volt, quindi utilizzeremo i 4,5 volt che si possono ottenere facilmente con il portabatterie.

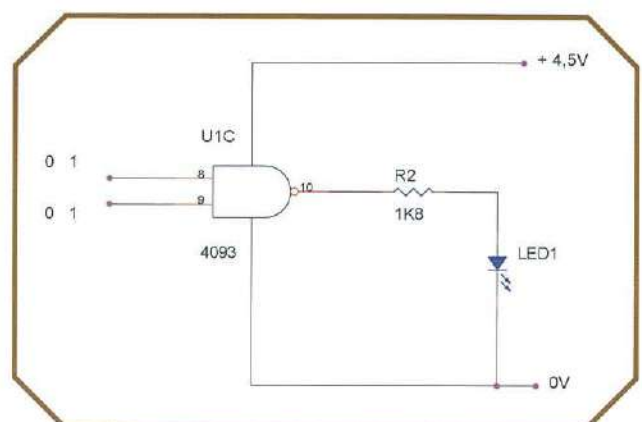
Non dobbiamo dimenticarci di collegare l'alimentazione del circuito integrato: il terminale di alimentazione negativo è sul 7 e quello positivo sul 14.

### Esperimento

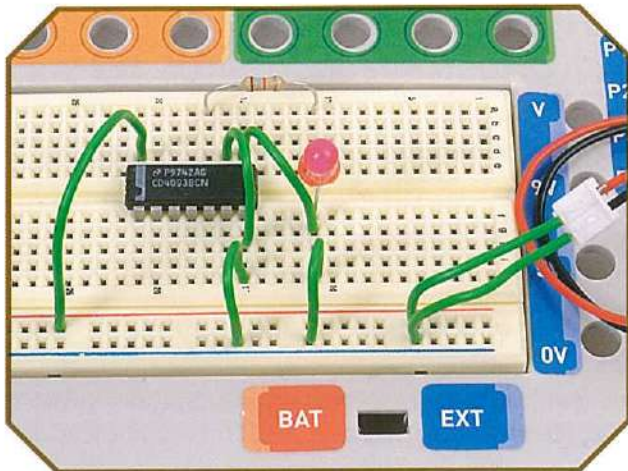
L'esperimento è molto semplice, basta applicare una combinazione di zeri e uno agli ingressi, terminali 8 e 9, per ottenere l'uscita. Le combinazioni 00, 01 e 10 devono illuminare il LED, mentre con la 11 esso deve rimanere spento. Si può ripetere l'esperimento con ognuna delle porte, perché, anche se può sembrare poco interessante a livello speri-



L'integrato 4093 contiene 4 porte NAND a due ingressi.



Schema di prova del 4093.



Prova del 4093.

mentale, ci permette di verificare se il circuito integrato funziona o meno, testando il funzionamento di ognuna delle sue porte.

## Generatore di onda quadra

Il tipo di porte contenuto nel 4093 ci permette di realizzare degli oscillatori usandone solamente una di esse, sarà possibile anche costruire 4 oscillatori diversi, dato che disponiamo di 4 porte.

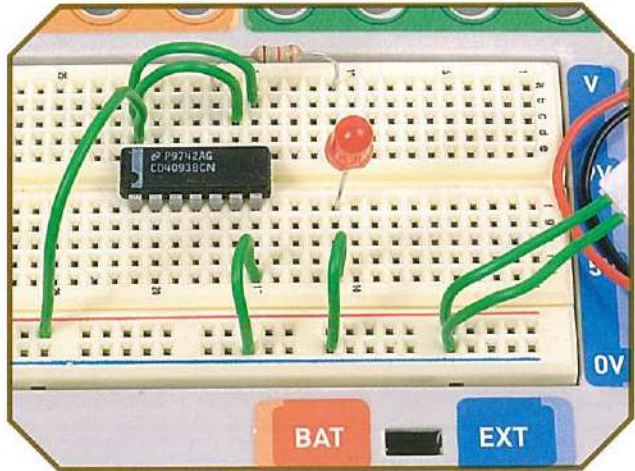
All'uscita dell'oscillatore si ottiene un'onda quadra. La frequenza di questo oscillatore dipende dai valori scelti per la resistenza R1 e per il condensatore C1, e si può calcolare in modo approssimativo mediante la formula:

$$f = 1/(0,7 \times R1 \times C1).$$

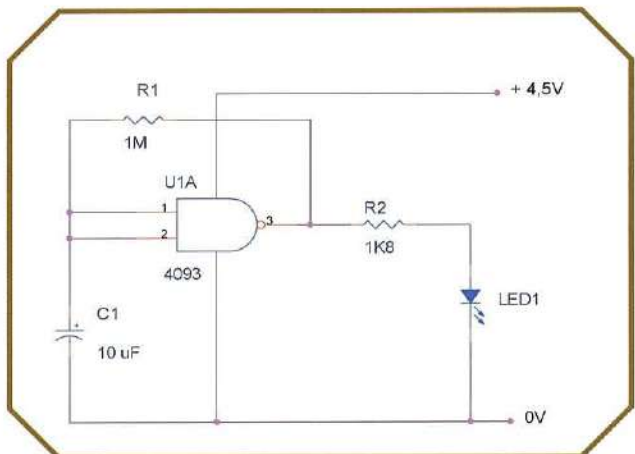
La resistenza R2 ha solamente il compito di limitare la corrente di polarizzazione del diodo LED che si utilizza come indicatore del funzionamento.

## Montaggio

Il montaggio è semplice, dobbiamo solamente scegliere una delle porte dell'integrato e collegare le due resistenze, il condensatore e il diodo LED. Prima di collegare l'alimentazione bisogna rivedere tutti i collegamenti; il negativo del condensatore elettrolitico deve essere collegato al negativo dell'alimentazione. L'alimentazione vi è già stata indicata in precedenza ed è stata collegata nell'ultimo esperimento.



Con i due ingressi a 1 il LED non si illumina.



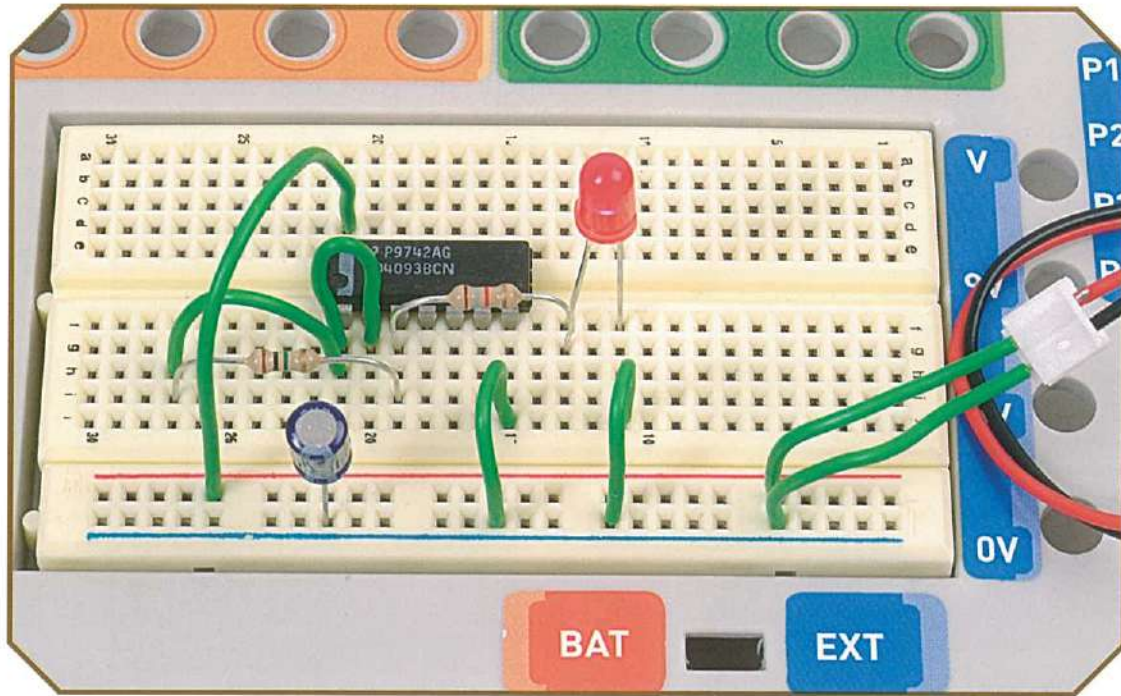
Astabile con 4093.

## Prove

Vi consigliamo di montare il primo circuito utilizzando i componenti dello schema, ovvero, R1 da 1 M e C1 da 1  $\mu$ F. La cadenza di illuminazione del LED è bassa, quindi impiega un po' di tempo per accendersi e spegnersi. Per aumentare il tempo potremmo collegare in parallelo con C1 l'altro condensatore da 10  $\mu$ F, mentre per diminuire il tempo possiamo fare al contrario, cioè abbassare la capacità.

Possiamo anche variare il valore della resistenza R1, e vi consigliamo di provare con i valori di 330 K e 47 K.

È possibile realizzare molte combinazioni di resistenze e condensatori per R1 e C1; ad esempio, con 47 K per R1 e 22 nF per C si ottiene una frequenza elevata, tanto da sembrare



Generatore astabile con 4093.

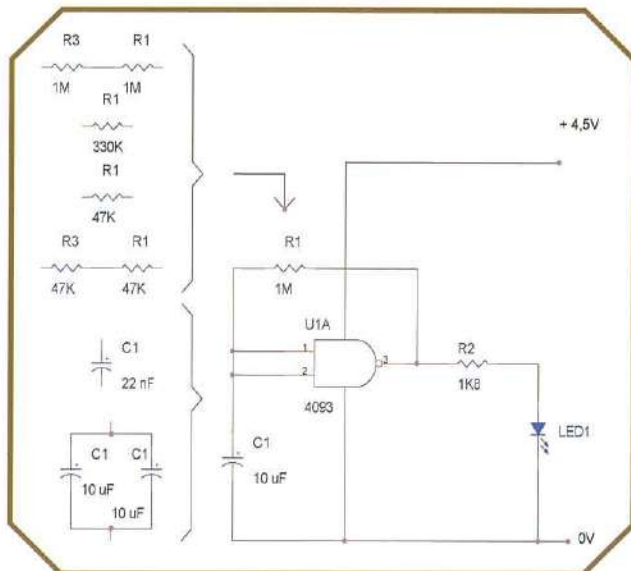
che il LED non si spenga, anche se il circuito sta funzionando, malgrado non lo si possa verificare perché l'occhio umano non può percepire variazioni di luce così rapide.

### Corrente di uscita

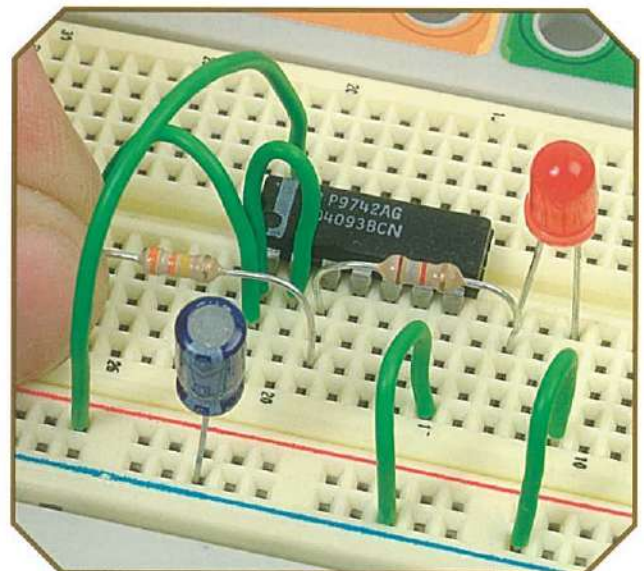
Con i circuiti elettronici è possibile realizzare molti esperimenti, i circuiti logici utilizzati in

elettronica digitale utilizzano correnti di uscita e di ingresso normalmente molto ridotte, dal momento che hanno il compito di mantenere un livello di tensione da applicare all'ingresso di un circuito ad alta impedenza, ovvero, che assorbe pochissima corrente.

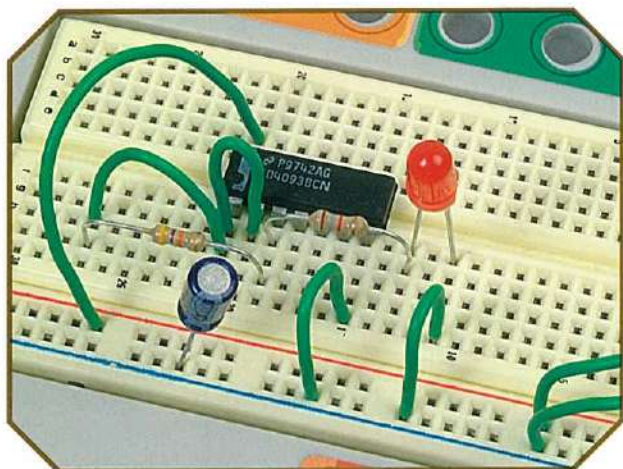
Il 4093 può fornire corrente sufficiente alla sua uscita, anche per alimentare un LED, tuttavia, se non vogliamo sovraccaricare le sue



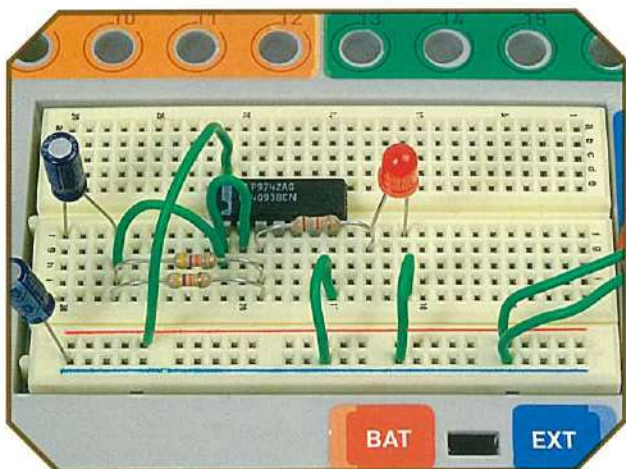
Schema.



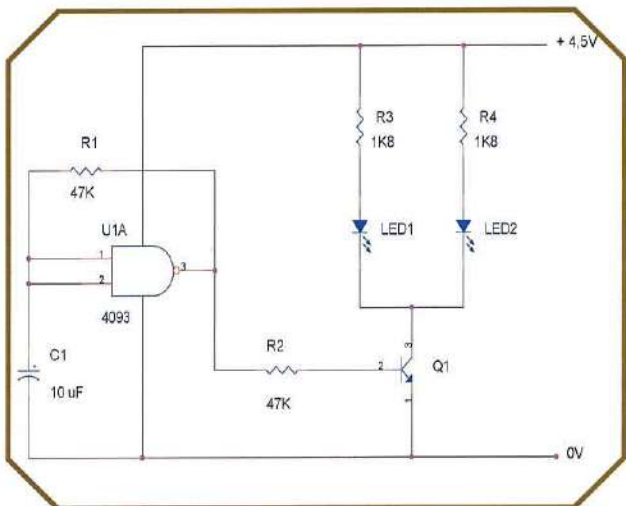
Aumento della frequenza sostituendo R1 con 330 K.



Prova con R1 da 47 K.



Si possono fare diverse combinazioni di componenti, provando anche i collegamenti in serie o in parallelo.



Schema dell'esperimento 3.

uscite o semplicemente se dobbiamo utilizzare un circuito integrato che, pur avendo un buon livello di uscita, non possa fornire corrente sufficiente, potremo modificare il circuito per fare in modo che controlli correnti più grandi alla sua uscita.

#### LISTA DEI MATERIALI

Circuito di base

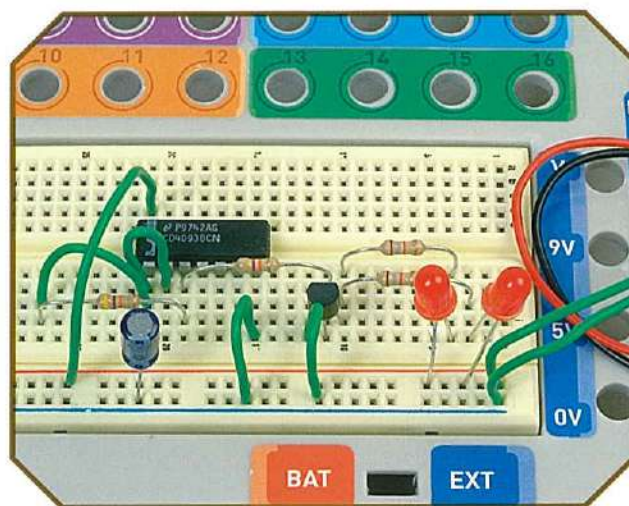
U1 Circuito integrato 4093  
 R1 Resistenza 1 M (marrone, nero, verde)  
 R2 Resistenza 1K8 (marrone, grigio, rosso)  
 C1 Condensatore 10 µF elettrolitico  
 LED1 Diodo LED rosso 5 mm

Per esperimento 3

U1 Circuito integrato 4093  
 R1, R2 Resistenza 47 K (giallo, viola, arancio)  
 R3, R4 Resistenza 1K8 (marrone, grigio, rosso)  
 C1 Condensatore 10 µF elettrolitico  
 Q1 Transistor NPN BC547 o BC548  
 LED1, LED2 Diodo LED rosso 5 mm

### Esperimento 3

Se osserviamo lo schema vedremo un oscillatore astabile costruito con una porta NAND del 4093 e una resistenza da 47 K alla sua uscita che porta direttamente il segnale alla base di un transistor Q1. Quando l'uscita di questo circuito integrato passa a livello alto il transistor conduce producendo l'illuminazione dei LED. Questo transistor può condurre senza problemi 100 mA. In questo caso viene utilizzato per illuminare due LED contemporaneamente.



Con un transistor si controlla la corrente di uscita.



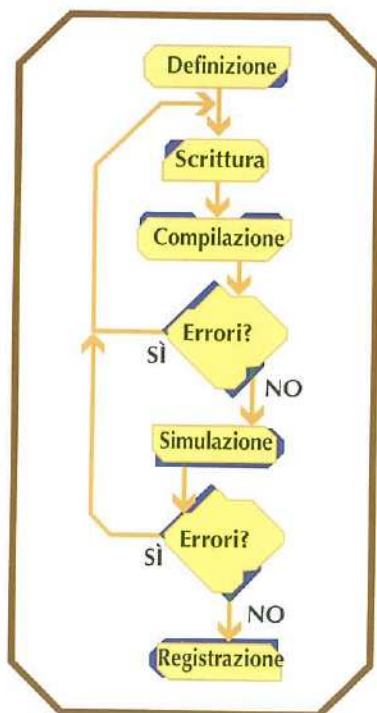
## Simulazione con MPLAB (I)

**N**ei capitoli precedenti abbiamo completato un programma e lo abbiamo liberato dagli errori: questo ha generato un file con estensione ".hex" che verrà scritto sul microcontroller, però rimangono ancora alcune cose importanti da fare. Il programmatore, prima di scrivere il programma sul PIC, deve assicurarsi che il codice creato corrisponda ai requisiti del progetto, ovvero che il suo programma faccia ciò che realmente si desidera. MPLAB offre la possibilità di simulare il programma, il suo potente simulatore ci permetterà di analizzare come funzionerebbe il programma se stesse girando su un PIC.

### Prima di simulare

Dobbiamo aprire MPLAB e caricare il progetto con cui stiamo lavorando. Dopo averlo aperto lo compileremo nuovamente per assicurarci che il nostro programma non contenga errori. Apparirà una videata per indicarci che la compilazione si è realizzata con successo, ovvero è stato creato un file con estensione ".hex" che può già essere scritto sul microcontroller.

Prima di scrivere il programma sul microcontroller è conveniente verificare che corrisponda realmente ai requisiti dell'applicazione, utilizzando il simulatore.

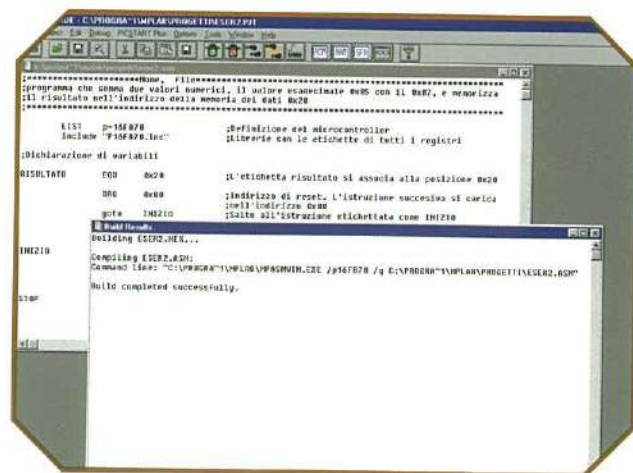


Fasi del progetto di un programma.

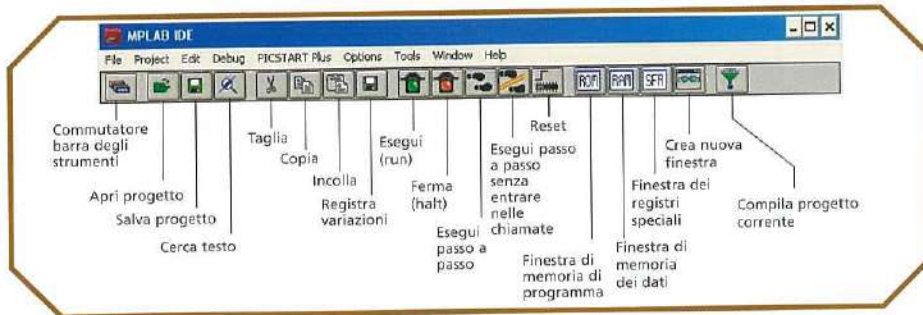
### La barra degli strumenti di simulazione

Se commutiamo tra le diverse barre degli strumenti, arriveremo a una identica, quella della figura alla pagina successiva. Benché ogni barra abbia una sua utilità specifica, possiamo vedere che alcune delle funzioni si ripetono sulle varie barre. Questo si fa per non dover sempre commutare tra le barre ogni volta che avremo bisogno di eseguire alcune azioni che esulano dall'utilità specifica propria della barra, ovvero, in ogni barra avremo le funzioni specifiche della barra stessa più alcune funzioni generali, in modo da semplificare la gestione del software.

Nella barra di simulazione, quindi, potremo vedere che le prime otto funzioni sono generali e le successive proprie della simulazione. L'ultima delle funzioni è quella di compilazione, dato che nella fase di simulazione potre-



Videata dove viene indicato il risultato della compilazione.



Barra degli strumenti di simulazione.

mo modificare il programma, però prima di simularlo nuovamente sarà necessario tornare a compilarlo.

## Finestre o window

Le finestre permettono di vedere cosa fa il nostro programma, come funziona internamente al PIC mano a mano che si sta eseguendo. Quando attiviamo l'opzione window, nella barra appare il menù a tendina della figura sottostante.

Program Memory  
Trace Memory  
EEPROM Memory  
Absolute Listing  
Map File

---

Stack  
File Registers  
Special Function Registers  
Show Symbol List...      Ctrl+F8  
Stopwatch...

---

Project

---

Watch Windows

---

Modify...

---

Tile Horizontal  
Tile Vertical  
Cascade  
Iconize All  
Arrange Icons

---

1 c:\archiv~1\mplab\proyec~1\ejer2.asm  
✓ 2 Build Results

Menù a tendina dell'opzione window della barra dei menù.

## Program memory

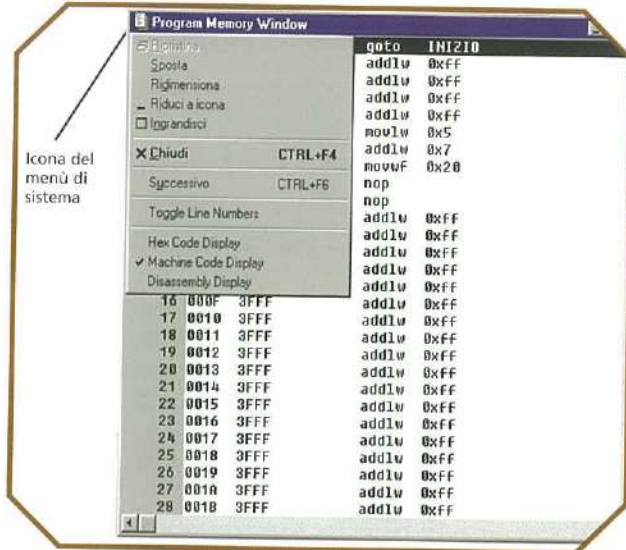
Selezionando questa opzione appare la videata riportata nella figura sottostante in cui è possibile vedere le posizioni della memoria occupate da ogni istruzione, il codice delle operazioni di ogni istruzione e la posizione di memoria che è stata assegnata a ogni etichetta.

Se clicchiamo con il mouse sull'icona nell'angolo superiore sinistro di questa finestra, la barra di menù del sistema, appare il menù a tendina della figura in cui potremo selezionare tre modi di vedere la memoria di programma:

Hex Code Display: rappresenta la memoria di programma con i dati in formato esadecimale. Questa opzione è utile quando vogliamo verificare se i dati sul microcontroller sono stati scritti correttamente, confrontando la videata del software di scrittura utilizzato se abbiamo scelto di scrivere in questo formato.

Program Memory Window			
	Address	Hex	Code
1	0000	2805	goto INICIO
2	0001	3FFF	addlw 0xFF
3	0002	3FFF	addlw 0xFF
4	0003	3FFF	addlw 0xFF
5	0004	3FFF	addlw 0xFF
6	0005	3005	movlw 0x5
7	0006	3E07	addlw 0x7
8	0007	00A0	movwf 0x20
9	0008	0000	STOP nop
10	0009	0000	nop
11	000A	3FFF	addlw 0xFF
12	000B	3FFF	addlw 0xFF
13	000C	3FFF	addlw 0xFF
14	000D	3FFF	addlw 0xFF
15	000E	3FFF	addlw 0xFF
16	000F	3FFF	addlw 0xFF
17	0010	3FFF	addlw 0xFF
18	0011	3FFF	addlw 0xFF
19	0012	3FFF	addlw 0xFF
20	0013	3FFF	addlw 0xFF
21	0014	3FFF	addlw 0xFF
22	0015	3FFF	addlw 0xFF
23	0016	3FFF	addlw 0xFF
24	0017	3FFF	addlw 0xFF
25	0018	3FFF	addlw 0xFF
26	0019	3FFF	addlw 0xFF
27	001A	3FFF	addlw 0xFF
28	001B	3FFF	addlw 0xFF

Videata della finestra di memoria di programma.



Apertura delle opzioni del menù di sistema.

**Machine Code Display:** questa opzione è quella predeterminata nel software e presenta il codice macchina assemblato così come abbiamo visto in precedenza.

**Disassembly Display:** espande il codice esadecimale disassemblando con i simboli, quindi risulta più facile seguire il programma.

Nei due modi precedenti, Machine Code e

Disassembly, l'istruzione a cui punta il contatore di programma si trova evidenziata.

### Trace memory

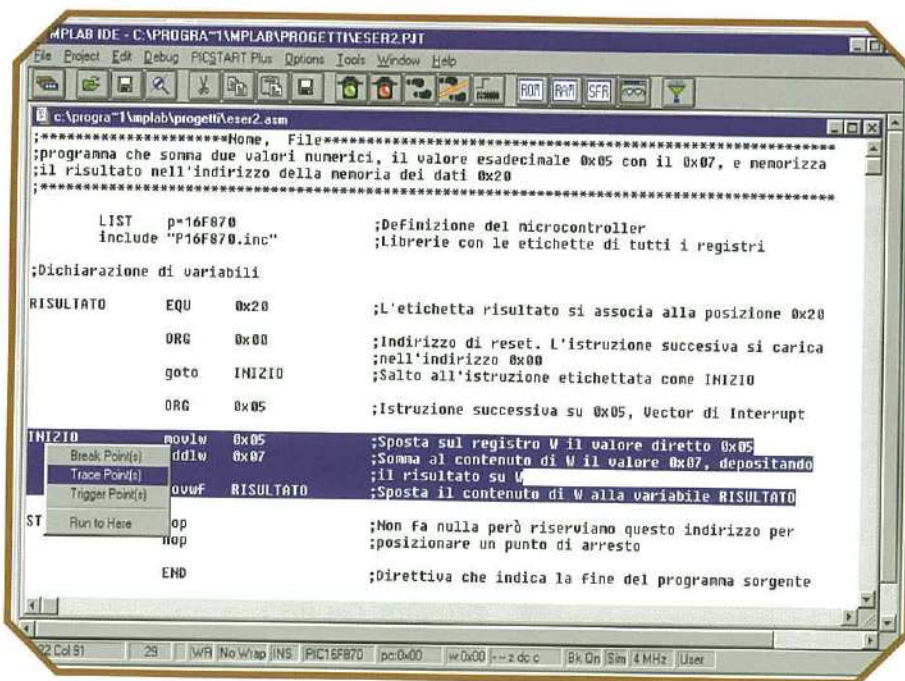
Questa finestra si chiama traccia di memoria e la sua funzione è quella di catturare in un preciso momento l'evoluzione del programma quando questo sta girando in tempo reale. Alcuni problemi si evidenziano solamente quando il programma sta girando e non durante l'esecuzione passo a passo, quindi, per rilevarli, utilizzeremo questo strumento. Mediante



Traccia di memoria ottenuta con le linee selezionate.

questa finestra potremo visualizzare un registro durante l'esecuzione del programma, registrando gli stati che assume per una successiva analisi. Prima di attivare questa opzione, evidenzieremo con il mouse le linee di codice delle quali vogliamo ottenere i dati durante l'esecuzione del programma e cliccheremo il pulsante destro del mouse. Apparirà il menù a tendina della figura.

Se selezioniamo Trace Point le linee selezionate verranno evidenziate in colore verde. Per eseguire il programma selezioneremo il semaforo verde dalla barra degli strumenti o sul menù di controllo Debug→Run→Run. Se dopo qualche secondo (il simulatore emula il funzionamento del PIC, ma è molto più lento di quest'ultimo) attiveremo l'icona del semaforo rosso oppure sul menù di controllo Debug→Run→Halt, fermeremo l'es-



Selezione delle linee di programma da caricare nella finestra di tracciamento.



Finestra della memoria EEPROM.

cuzione del programma. Se ora attiveremo l'opzione Trace Memory potremo vedere la traccia ottenuta. Nella figura si può vedere la traccia ottenuta per il nostro esempio. Il simulatore mostra in questa finestra il valore del tempo impiegato nell'eseguire ogni linea di programma, oltre che qualsiasi variazione prodotta sui registri dall'esecuzione del codice di istruzione.

## EEPROM Memory

Questa finestra è utile solamente per i PIC che dispongono di tale dispositivo. Il contenuto di questa memoria si può vedere selezionando l'opzione Window→EEPROM Memory e appare come nella figura in alto. Il contenuto della memoria non può essere modificato da questa finestra, ma grazie a un'altra opzione del menù è possibile eseguire delle modifiche.

## Absolute Listing

Se selezioniamo questa finestra appare il file del nostro progetto con estensione ".lst". Durante la compilazione si crea un file con lo



Finestra dello stack.

stesso nome del nostro programma sorgente ma con questa estensione e selezionando la finestra Absolute Listing possiamo editare il file. In esso troviamo il codice sorgente in modo assoluto e il codice oggetto generato, così come possiamo vedere nella figura.

Al termine di questo file troveremo l'informazione delle etichette utilizzate nel programma, su quale linea si trovano, la memoria utilizzata, la memoria libera e gli errori, warning e messaggi generati dal compilatore.

## Stack

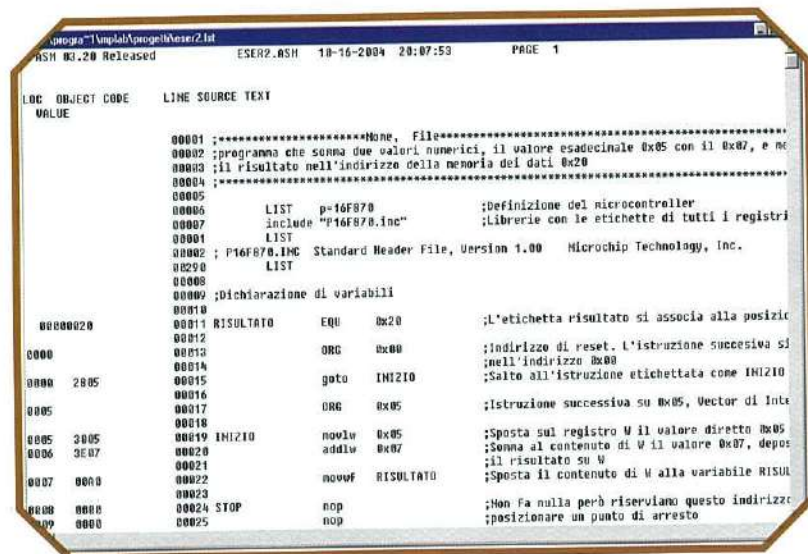
Grazie a questa opzione potremo visualizzare il contenuto dello stack. Il contenuto può essere mostrato con o senza il numero di linea. Sceglieremo il formato di presentazione cliccando sull'icona nella parte superiore sinistra della finestra.

Se il contenuto eccede oltre la dimensione dello stack MPLAB lo indica mostrando il messaggio "underflow".

## Conclusioni

La simulazione non avrebbe senso se non si potesse utilizzare il funzionamento interno del PIC e, per questa ragione, è necessario vedere tutte le finestre che ci offre MPLAB. Conoscendo queste e imparando bene la gestione del simulatore sarà più facile mettere a punto il programma per una sua successiva scrittura sul microcontroller. Ma questo lo vedremo nei prossimi capitoli.

File "eser2.lst"  
visibile selezionando Absolute Listing.







## Bistabile RS

**D**isponiamo già dei componenti necessari per costruire un bistabile RS. Faremo delle prove con due circuiti, uno più semplice che corrisponde al circuito teorico ma che presenta qualche problema di funzionamento, e l'altro più elaborato, che risolve alcuni dei problemi ed è di utilizzo pratico.

### Bistabile RS

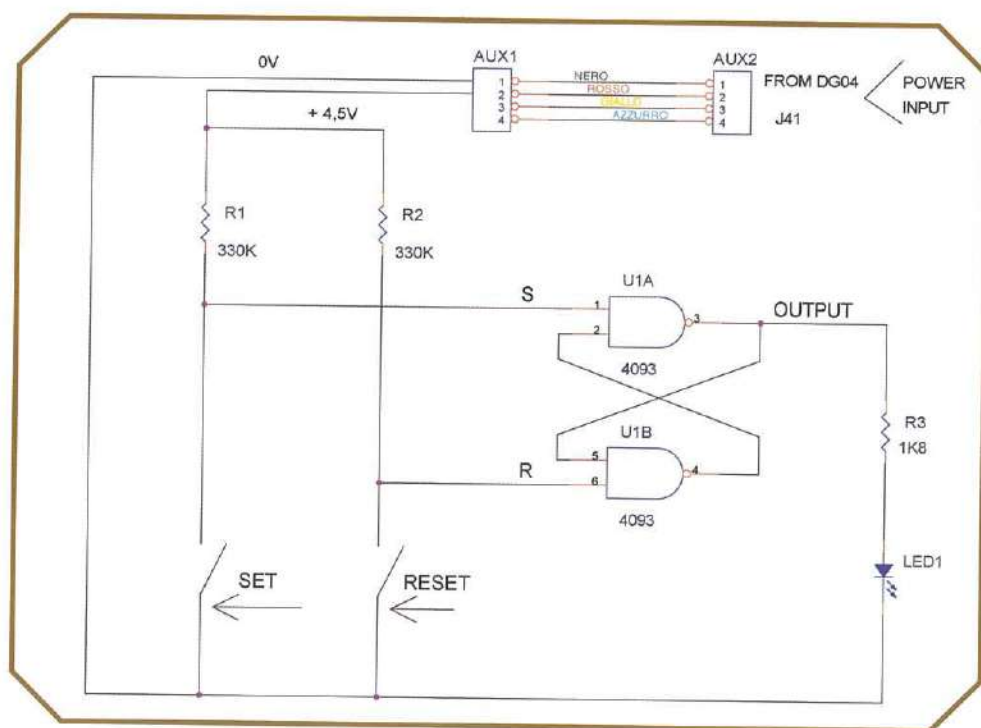
Anche se ne abbiamo già parlato in precedenza, ricordiamo che questo circuito ha due ingressi, uno identificato come S da SET, che è quello di attivazione, e l'altro chiamato R da RESET, che è quello di disattivazione. Il nostro circuito di base corrisponde ai terminali 1 e 6 del circuito integrato 4093.

Questo tipo di circuiti ha due uscite che, nel nostro caso, sono collegate ai terminali 3 e 4, anche se noi utilizzeremo solamente la 3, a cui verrà collegato un LED indicatore con la relativa resistenza di limitazione. Lo scopo di questo LED è quello di sapere in quale stato si trova l'uscita del bistabile, a livello alto il LED si illumina, mentre per il livello basso rimane spento. Gli ingressi si mantengono a livello alto utilizzando resistenze da 330 K.

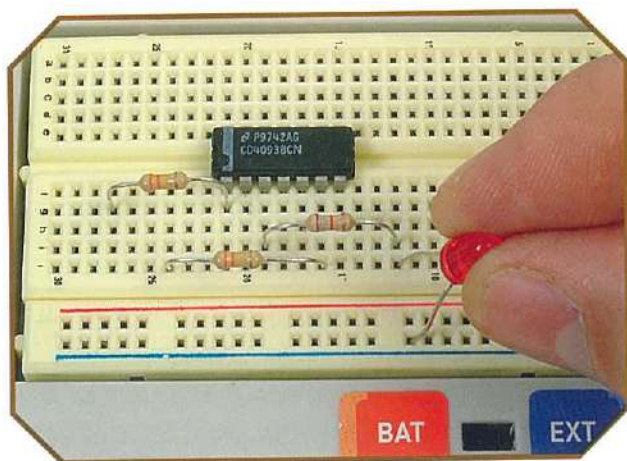
### Montaggio

Il montaggio del circuito del bistabile più semplice si realizza con la procedura abituale, facendo attenzione alla polarità del LED, a quella dell'alimentazione e all'orientamento del circuito integrato. Le fotografie sono un buon aiuto, anche se il montaggio si deve eseguire seguendo lo schema elettrico.

Questo circuito e quello migliorato, che descriveremo di seguito, si possono alimentare tramite la scheda DG04 fino a quando il laboratorio non sarà montato e potremo ottenere l'alimentazione dalle molle di collegamento siglate come 0 e 5 V. Alimentando tramite la scheda DG04, così come la stiamo utilizzando, con i collegamenti ausiliari del portabatterie, non è necessario collegare i



Schema elettrico del bistabile di base realizzato con porte NAND.

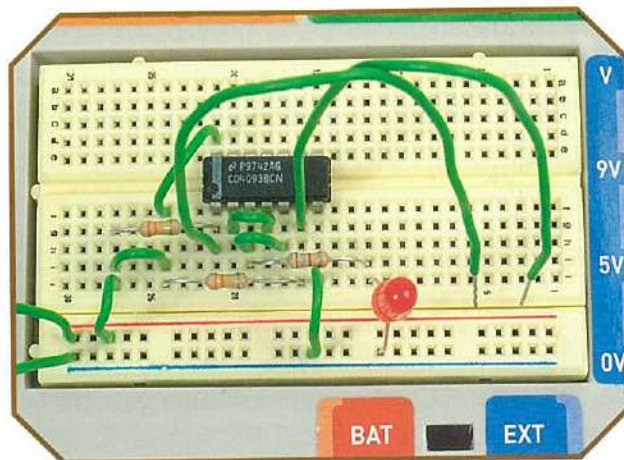


Componenti del bistabile di base.

ponticelli della scheda DG04. Nel circuito integrato 4093 il terminale di alimentazione negativo è il 7, mentre quello positivo è il 14, e questi terminali, solitamente, non vengono inseriti negli schemi. L'ideale sarebbe disporre di due pulsanti per realizzare i collegamenti di SET e RESET, tuttavia è molto facile sostituirli con due fili i cui capi verranno collegati temporaneamente al negativo dell'alimentazione per realizzare le funzioni di SET e RESET.

## Esperimento

Dopo aver montato il circuito il LED si potrebbe accendere oppure no, a seconda di quale porta condurrà per prima. Se eseguiamo temporaneamente il collegamento indicato come SET, l'uscita del circuito passerà a livello alto e



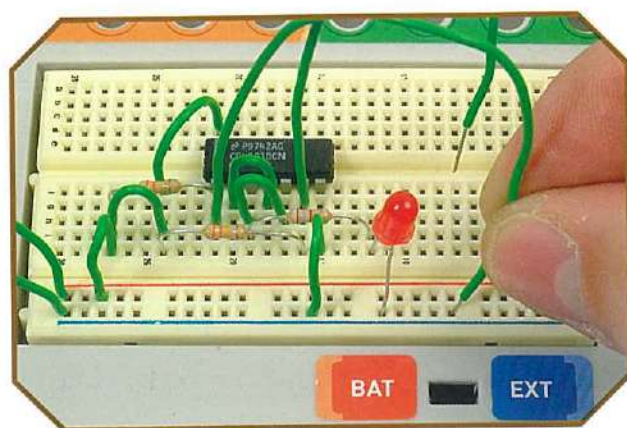
Bistabile di base.

il LED si illuminerà e rimarrà tale, fino a quando non collegheremo temporaneamente il collegamento indicato come RESET, questo farà passare l'uscita allo stato basso e il LED si spegnerà.

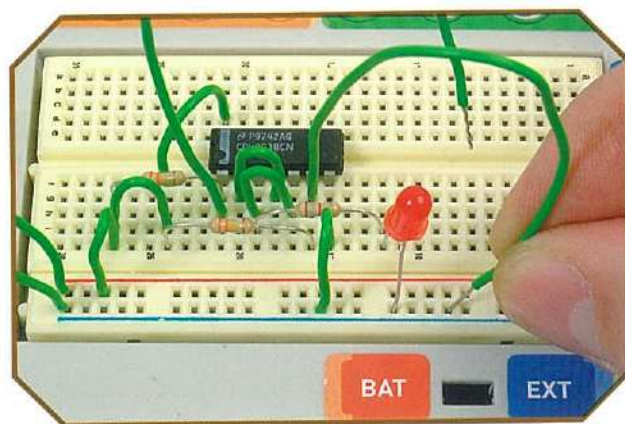
Come possiamo vedere, ciò che in realtà stiamo facendo è memorizzare uno stato di uscita.

Non dobbiamo mai attivare contemporaneamente entrambi gli ingressi.

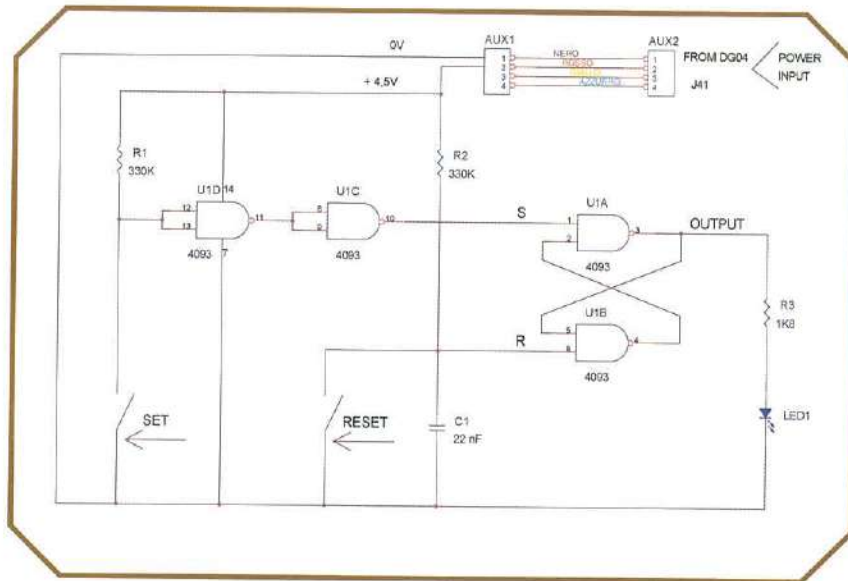
Quando il laboratorio sarà più completo potremo ripetere questo esperimento utilizzando i pulsanti per i collegamenti di SET e RESET. Questo circuito può presentare alcuni problemi di funzionamento dovuti alle imperfezioni dei contatti, inoltre non sappiamo in quale stato si troverà al momento di collegare l'alimentazione. Realizziamo ora alcune migliorie nel circuito.



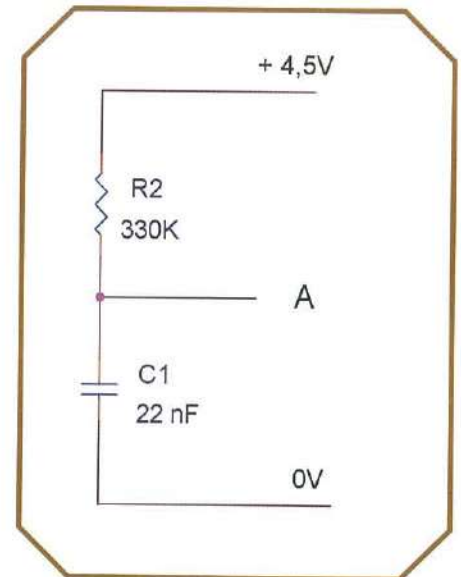
Collegamento temporaneo del filo di SET.



Collegamento del RESET.



Schema elettrico del bistabile migliorato.



Dettaglio del circuito di RESET all'accensione.

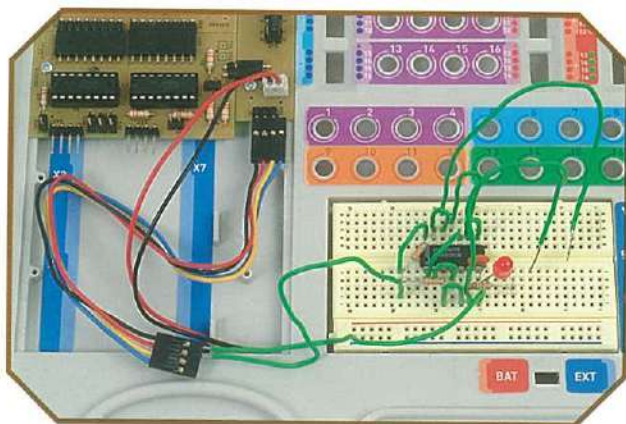
## Miglioramenti

Se osserviamo lo schema del bistabile migliorato potremo vedere che all'ingresso di RESET è stato aggiunto un condensatore C1.

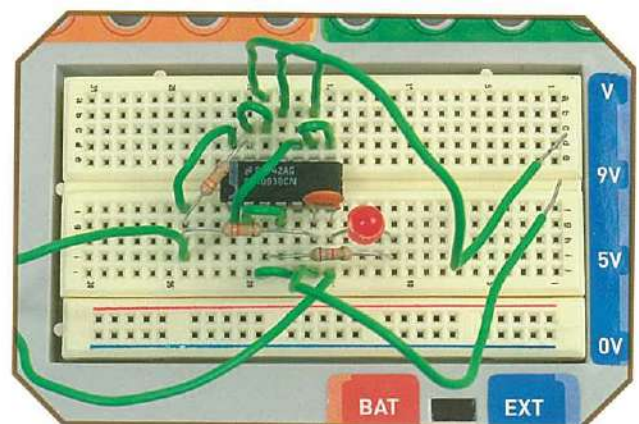
Il compito di questo condensatore è quello di garantire lo stato iniziale di RESET, dato che, al momento di collegare l'alimentazione, provoca uno zero temporaneo perché la carica del condensatore non è istantanea, quindi, collegando l'alimentazione, la tensione iniziale del condensatore è uguale a zero, il circuito è disattivato e il LED rimane

spento. La tensione in questo punto, terminale 6 del circuito integrato, passa a livello alto dopo un certo tempo e così si mantiene fino a quando si esegue il collegamento di RESET.

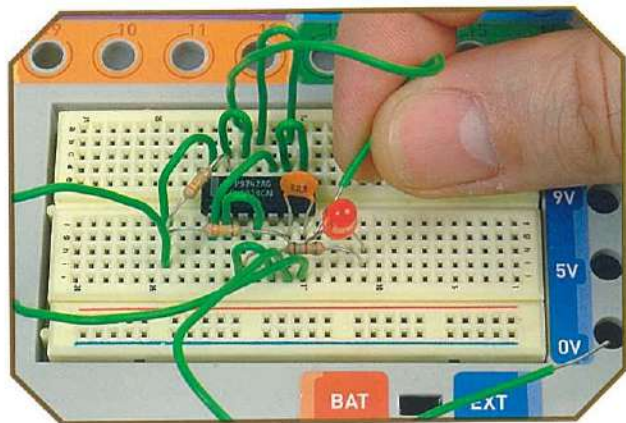
Un altro miglioramento è l'utilizzo di due porte all'ingresso di SET per eliminare i possibili rimbalzi dei contatti, stabilizzando, quindi, il funzionamento del circuito. Il funzionamento rimane lo stesso, eseguendo temporaneamente il collegamento di SET si illumina il LED e, facendo la stessa cosa con l'altro collegamento, il LED si spegne.



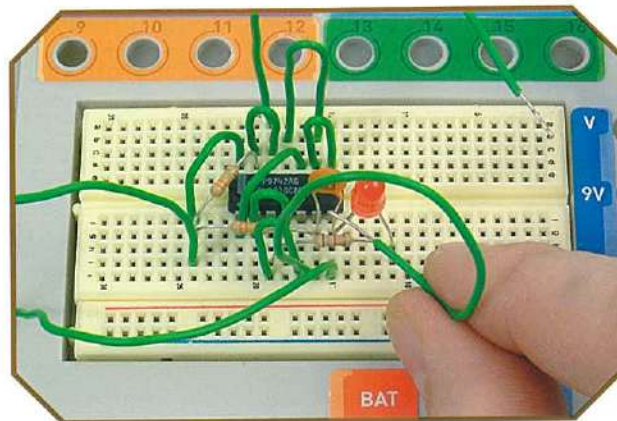
Bistabile migliorato montato sulla Bread Board.



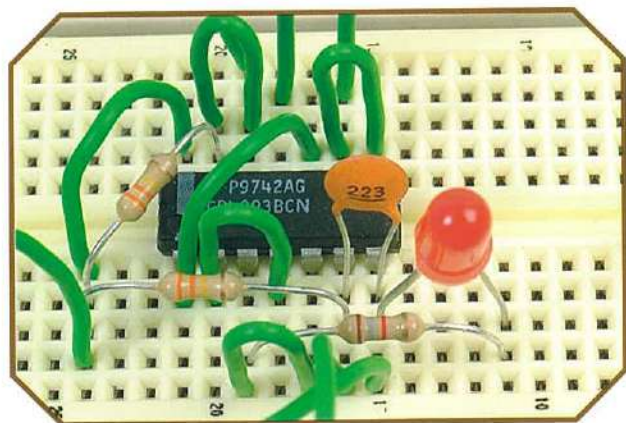
Bistabile migliorato che comprende il condensatore C1.



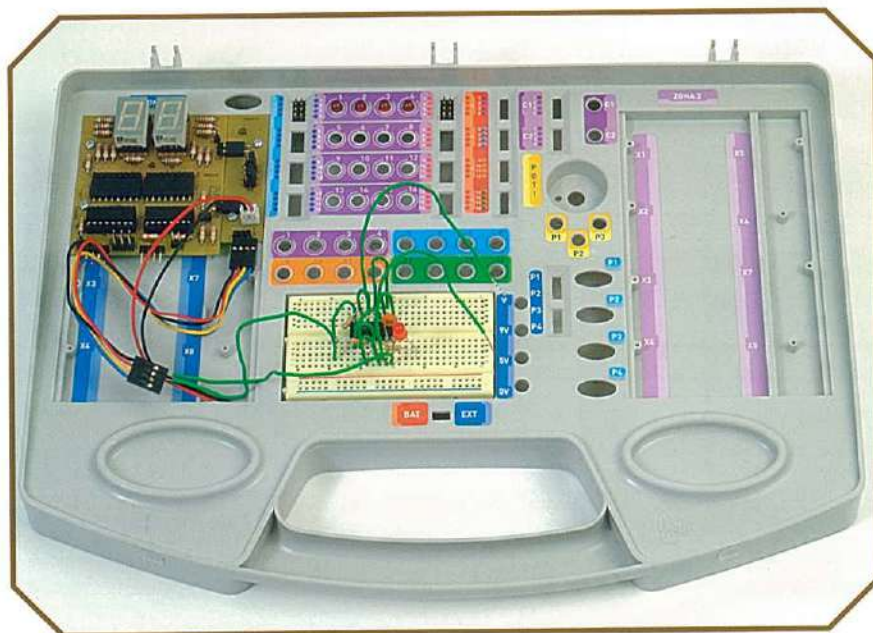
Collegamento di SET, si tocca qualsiasi punto del negativo.



Collegamento di RESET, si tocca il terminale negativo del LED.



Dettaglio del condensatore C1.



Aspetto del laboratorio con l'esperimento del bistabile.

### Conclusioni

Questo circuito permette di memorizzare uno stato o l'altro in base al pulsante premuto, e questo stato si mantiene fino a quando dura l'alimentazione.

È stato anche verificato il modo di utilizzare un condensatore per forzare lo stato iniziale del bistabile.

#### LISTA DEI MATERIALI

- |                     |   |
|---------------------|---|
| Circuito di base    |   |
| U1                  | Circuito integrato 4093                     |
| R1, R2              | Resistenze 330 K (arancio, arancio, giallo) |
| R3                  | Resistenza 1K8 (marrone, grigio, rosso)     |
| LED1                | Diode LED rosso 5 mm                        |
| Circuito migliorato |   |
| U1                  | Circuito integrato 4093                     |
| R1, R2              | Resistenze 330 K (arancio, arancio, giallo) |
| R3                  | Resistenza 1K8 (marrone, grigio, rosso)     |
| C1                  | Condensatore 22 nF                          |
| LED1                | Diode LED rosso 5 mm                        |



# Simulazione con MPLAB (II): finestre

**C**ontinuiamo l'analisi delle possibilità che ci offre MPLAB quando si simula il funzionamento di un programma. Abbiamo visto quali sono le finestre di simulazione, ma ci rimangono ancora da studiare alcune di esse che risultano fondamentali in questo processo.

## File Registers

Quando nella barra dei menù selezioniamo Window, appare un menù a tendina dove di seguito alle cose già viste, apparirà l'opzione File Registers. In questa finestra MPLAB presenta una lista di tutti i registri di utilizzo generale (GPR) del microcontroller. Come succedeva con alcune delle finestre viste in precedenza, ci sono tre modi di visualizzazione che si possono selezionare con l'icona situata nell'angolo superiore sinistro.

**Hex Display:** questa opzione visualizza i registri con i dati in formato esadecimale, così come li possiamo vedere nella figura riprodotta qui sotto.

**Symbolic Display:** questo formato presenta i registri di utilizzo generale con le loro etichette, se le hanno, e il loro contenuto in esa-

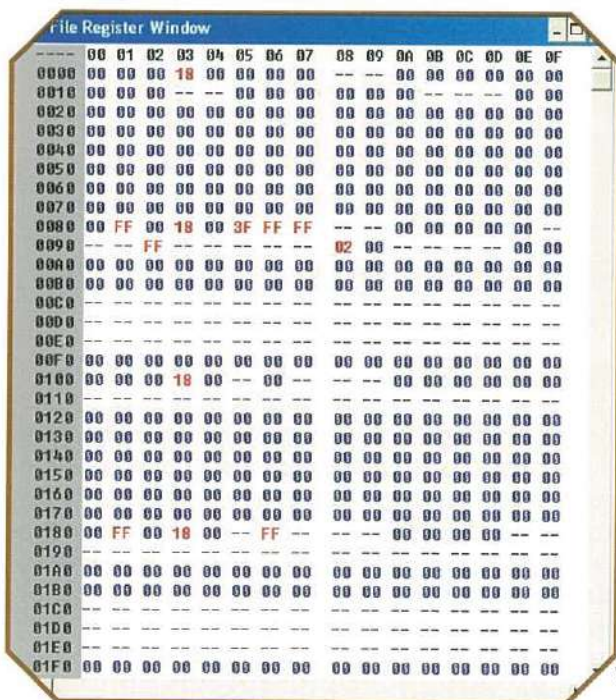
decimale, decimale, binario e formato carattere.

È la visualizzazione più pratica perché oltre alla posizione nella memoria possiamo identificare facilmente sia il registro che il suo contenuto.

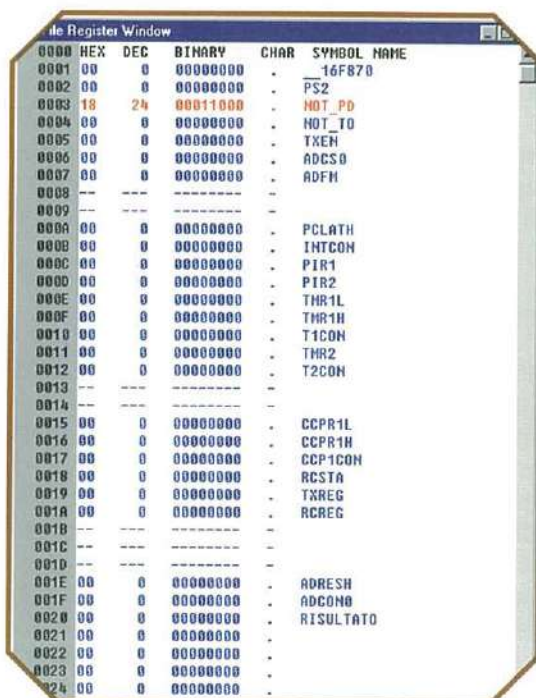
**ASCII Display:** in questo caso i dati dei registri sono presentati in codice ASCII.

Tramite questa finestra potremo modificare il contenuto di uno o più registri. Se puntiamo con il mouse sul primo registro che vogliamo modificare e lo selezioniamo mediante il pulsante destro, si attiverà l'opzione Fill Register(s). Questa opzione ci permetterà di modificare il valore del registro; se con il mouse selezioniamo tutto un blocco e clicchiamo il pulsante destro possiamo selezionare diversi registri.

Più avanti commenteremo questa comoda opzione.



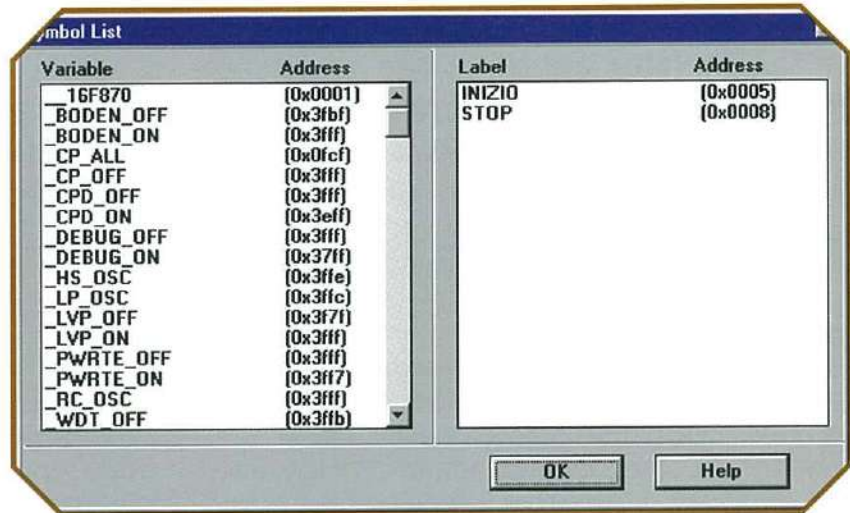
Listato dei registri di utilizzo generale del sistema.



Registri di utilizzo generale nel modo visualizzazione Symbolic.



I registri di utilizzo generale in modo ASCII.



Listato dei simboli utilizzati nel programma.

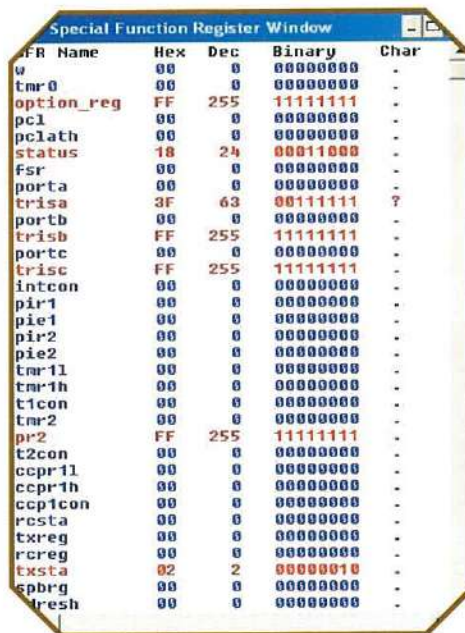
## Special Function Registers

Selezionando Window → Special Function Registers potremo vedere il contenuto dei registri con funzioni speciali (FSR). Durante la simulazione questa è una delle finestre da tenere normalmente aperta, il formato fornito da questa finestra risulta molto utile per analizzare il contenuto di questi registri in ogni momento. Come mostra la figura, in questa finestra appaiono tutti i registri FSRs identificati con il loro nome e con il loro contenuto nei formati esadecimale, decimale, binario e ASCII.

Per modificare il contenuto di uno di questi registri dobbiamo fare doppio clic sul registro prescelto e immediatamente apparirà la finestra Modify, che permetterà la scrittura del registro.

## Show Symbol List

In questa finestra sono riportati i simboli delle variabili e delle etichette, utilizzati nel codice sorgente del nostro programma. Dopo aver compilato il programma si genera un file con estensione ".cod" che contiene la definizione di tutti i simboli utilizzati nel programma. Questa finestra ha una funzione informativa riguardo ai simboli che abbiamo utilizzato e serve unicamente come orientamento in modo da poter identificare gli indirizzi di memoria a cui sono associati questi simboli.



Finestra dei registri di funzione speciali (FSRs).

## Stopwatch

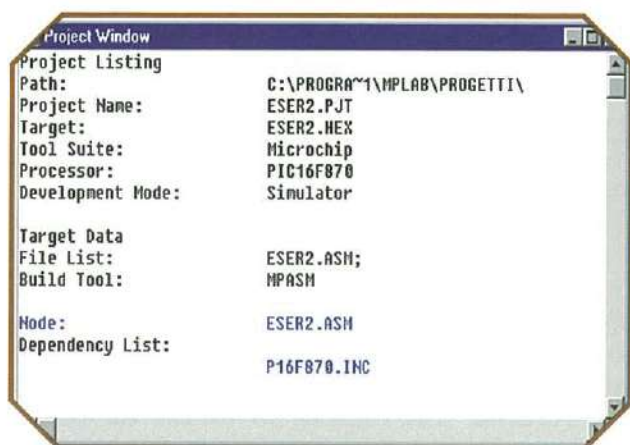
Durante la simulazione spesso ci tornerà utile poter calcolare il tempo di esecuzione di un programma o di una subroutine. Possiamo fare questo contando il numero di istruzioni che si realizzano e moltiplicandolo per quattro volte la frequenza del segnale di clock, o per otto nel caso in cui le istruzioni siano di salto, oppure lo potremo fare utilizzando il nostro software. MPLAB offre la possibilità di conta-



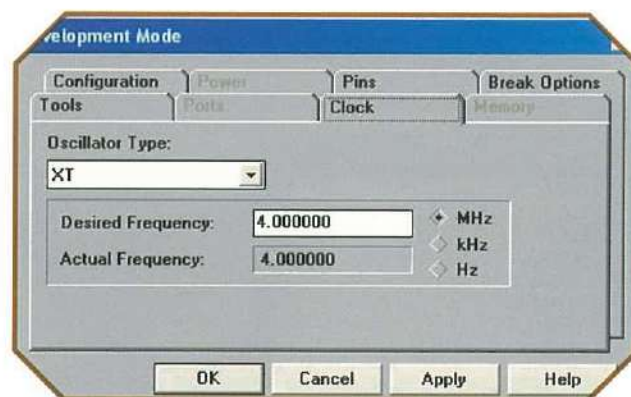
Finestra Stopwatch per contare il tempo di esecuzione delle istruzioni del nostro programma.

re il tempo di esecuzione delle istruzioni del programma mediante questo utile "cronometro", che calcola il tempo in base alla frequenza definita per il microcontroller che stiamo simulando. Per iniziare a utilizzare questo software è necessario definire la frequenza di lavoro per il microcontroller e questa impostazione rimarrà memorizzata anche per il resto dei progetti. Nel caso in cui si voglia modificare questa frequenza bisognerà eseguire l'operazione tramite il menù Option → Development Mode e dopo aver aperto questa finestra bisognerà selezionare la scheda Clock. Qui potremo scegliere il tipo di oscillatore da simulare e la frequenza di lavoro.

Possiamo mantenere aperta la finestra Stopwatch durante la simulazione per verificare i cicli e il tempo che impiega a essere eseguita ogni istruzione oppure il programma completo.



Finestra di progetto.



Tramite questa finestra è possibile cambiare la configurazione dell'oscillatore del microprocessore.

## Project Window

La finestra del progetto serve per presentare i dati di riferimento del progetto che abbiamo aperto. Essa ci indica l'indirizzo dove il progetto è memorizzato, i nomi dei file relativi e parte della configurazione del software per il progetto in questione. Se il progetto è stato assemblato o compilato, in questa finestra sono riportati tutti i file inclusi in esso. Se selezioneremo uno dei file riportati nella finestra, mediante un doppio clic del mouse, potremo aprire questo file per un'eventuale revisione.

Questa finestra ha una funzione informativa ed è raramente utilizzata durante il lavoro con MPLAB.

## Watch Window

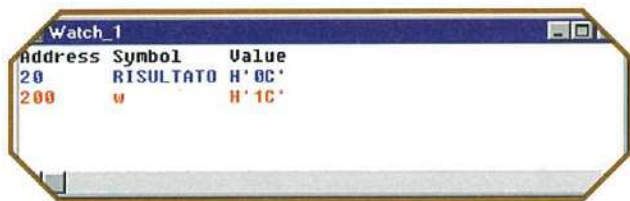
Selezionando questa opzione sul menù si apre un altro sottomenù dove potremo scegliere cinque opzioni.

New Watch Window: permette di creare una finestra temporanea per visionare i contenuti dei registri che desideriamo. Se durante la simulazione desideriamo vedere il contenuto di alcuni registri particolari, creeremo una finestra di questo tipo e selezioneremo i registri considerati importanti per l'analisi del programma.

Load Watch Window: le finestre create si possono salvare. Quando selezioniamo questa opzione stiamo informando il simulatore che desideriamo caricare una finestra precedentemente configurata, una finestra che abbiamo creato e che si trova memorizzata sull'Hard Disk del computer. Le finestre si memorizzano con estensione ".wat".



Mediante questa opzione possiamo aggiungere i registri che desideriamo analizzare in una finestra.



Aspetto che avrà la nostra finestra dopo aver aggiunto i registri.

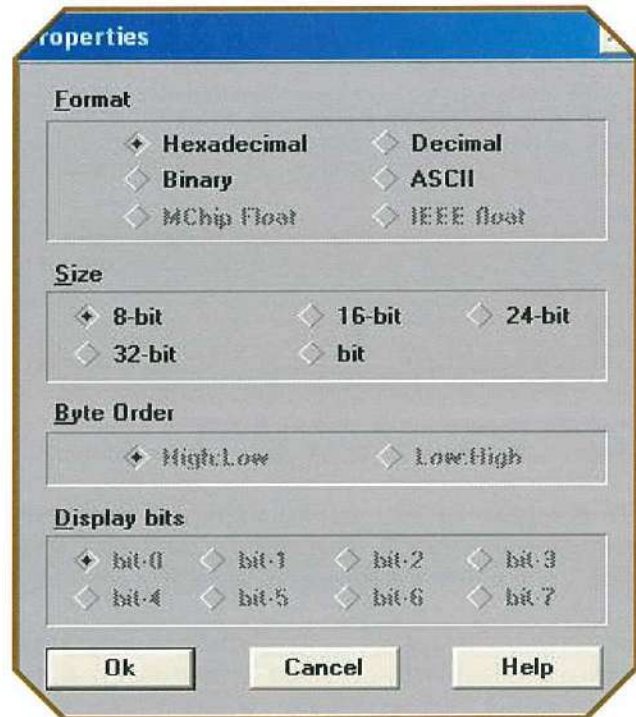
**Add to Active Watch:** questa opzione si abilita solamente se abbiamo già una finestra aperta e permette di aggiungere nuovi registri alla finestra che abbiamo creato.

**Edit Active Watch:** come la precedente anche questa opzione si attiva solamente quando abbiamo una finestra già aperta e permette di editare i registri che sono stati inseriti nella finestra creata.

**Save Active Watch:** mediante questa opzione possiamo salvare sull'Hard Disk la finestra che abbiamo creato; inoltre possiamo salvare la finestra quando dopo averla creata vogliamo chiuderla. Il software non ci chiederà al momento della chiusura se vogliamo salvare la finestra.

Per aggiungere registri alla finestra dobbiamo selezionare il registro e cliccare sull'opzione Add. Possiamo aggiungere tanti registri quanti ne desideriamo, o creare tante finestre quante ne consideriamo opportune. Normalmente i registri vengono raggruppati per funzioni comuni, all'interno di diverse finestre per la loro analisi.

Possiamo aggiungere dei registri mediante Add, però possiamo anche eliminare un regi-



Quadro di dialogo per modificare le proprietà della presentazione nella finestra.

stro che abbiamo aggiunto mediante l'opzione Delete. Inoltre potremo scegliere le proprietà della visualizzazione del registro mediante l'opzione Properties, o selezionando Edit Active Watch e facendo clic su Properties.

Come per la maggior parte delle finestre di simulazione, anche qui troviamo nell'angolo superiore sinistro l'icona del menù di controllo. Mediante questo menù possiamo aggiungere il numero di linee che desideriamo o inserire, cancellare o editare uno dei registri della finestra. Tramite questo menù è possibile inoltre salvare la finestra.

## Conclusioni

Termineremo lo studio delle finestre di simulazione analizzando l'opzione che permette di modificare il valore di un registro e vedendo anche le diverse visualizzazioni durante la simulazione. Avendo chiaro le opzioni di analisi che ci offre MPLAB tratteremo le opzioni specifiche della simulazione e infine lavoreremo con esempi in modo da poter sfruttare al massimo questo potente strumento.





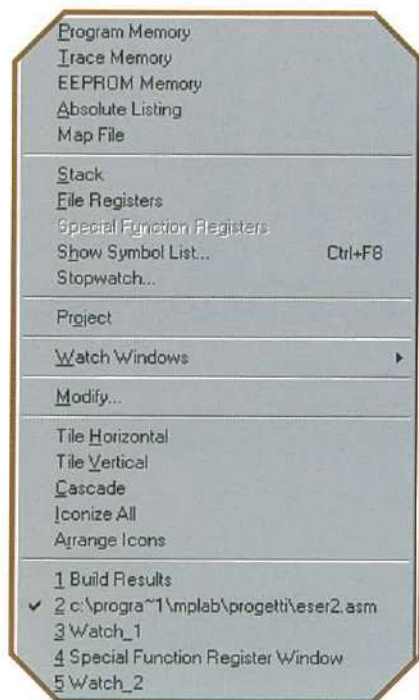
## Simulazione con MPLAB (III)

**T**ermineremo lo studio delle finestre di simulazione con l'opzione che permette di modificare il contenuto di un registro e le diverse visualizzazioni o presentazioni del display durante la simulazione. Siamo pronti per iniziare l'analisi delle diverse opzioni di simulazione di MPLAB.

### Finestra Modify

Quando attiviamo l'opzione Window → Modify il simulatore ci offre la possibilità di cambiare il contenuto di un registro. Per fare questo dobbiamo indicare al software l'indirizzo del registro da modificare, inserendolo tramite il campo Address e specificare il tipo di memoria in cui è contenuto (dati, programma, stack o EEPROM). Il valore che vogliamo dare al registro deve essere inserito nel campo Data/Opcode, indicando anche il formato del dato, esadecimale o decimale. Se clicchiamo su Write scriveremo sul registro il valore desiderato.

Questa opzione è molto comoda quando vogliamo simulare degli ingressi sul PIC, dato che potremo modificare il contenuto dei registri PORTA, PORTB e PORTC.



Opzioni del menù Window.

### Le finestre di simulazione

Abbiamo visto nel dettaglio tutte le finestre di simulazione che ci offre MPLAB. Nella simulazione possiamo visualizzare il contenuto di tutti i registri del nostro microcontroller e analizzare come si evolvono lungo il corso del programma.

Tutti i programmi che realizzeremo devono passare tramite il simulatore prima di essere scritti sul microcontroller, quindi dovremo prendere confidenza con la gestione di queste finestre.

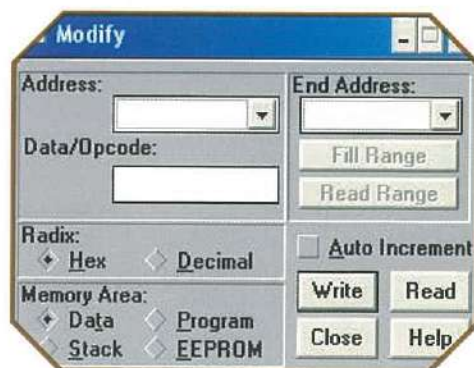
### Opzioni di presentazione

Un buon progettista terrà aperte diverse finestre durante la simulazione del programma. Dobbiamo evitare che la videata si converta in un caos, in un agglomerato di finestre senza alcun ordine, e a questo scopo le organizzeremo in modo che la loro analisi risulti comoda e semplice.

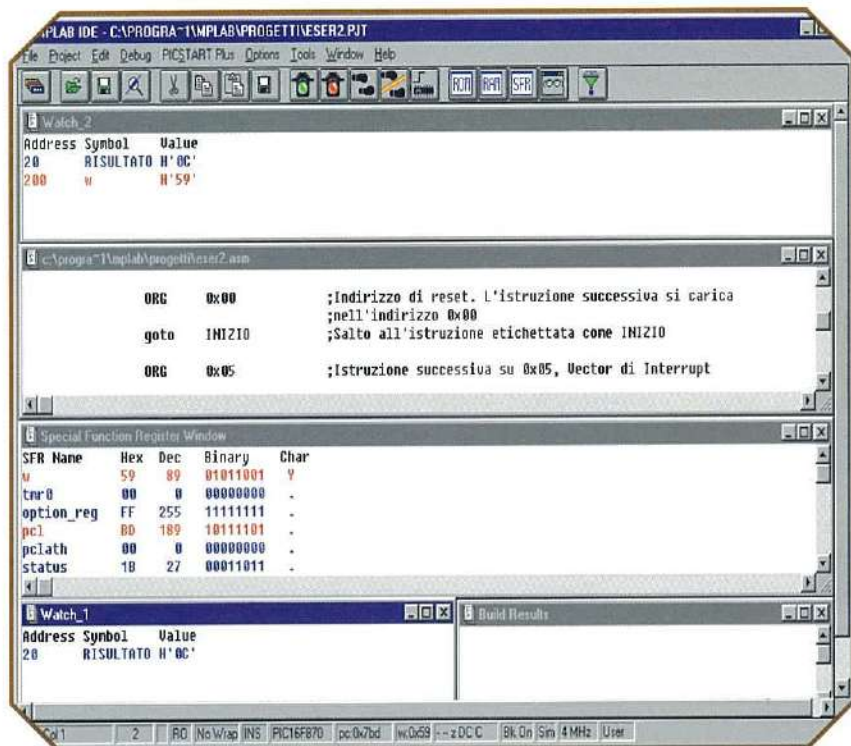
Se torniamo sul menù all'opzione Window vedremo che quando si apre appaiono le diverse scelte:

— Tile Horizontal: organizza le finestre in file, una sotto l'altra.

— Tile Vertical: organizza le finestre in colonne.



Il contenuto di un registro si può modificare mediante questa opzione.



Disposizione orizzontale delle finestre.

- Cascade: questa opzione presenta le finestre in cascata.
- Iconize All: per minimizzare tutte le finestre.
- Arrange Icons: ordina le finestre minimizzate.

Anche se il software ci offre questa scelta, saremo noi che posizioneremo le finestre in base alle nostre esigenze e al risultato funzionale. Un esempio dell'aspetto di MPLAB durante la simulazione è quello riportato in figura.

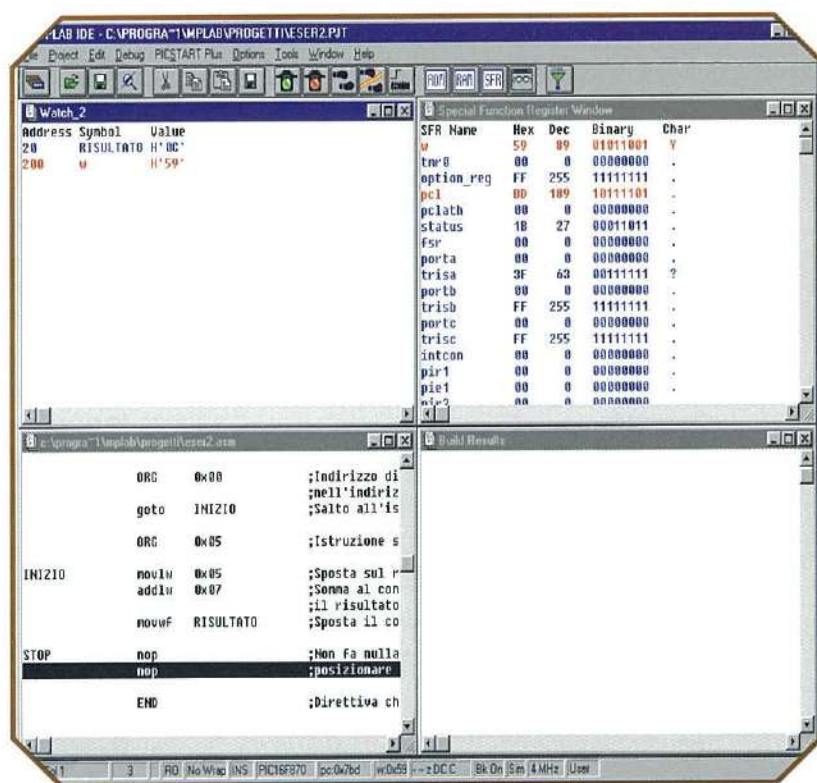
## Simulazione

Facendo un piccolo ripasso di ciò che abbiamo visto finora possiamo dire di aver imparato a scrivere un programma, a compilarlo, a togliere gli errori e sappiamo anche attivare tutti i dispositivi offerti dal software per sfruttare la simulazione, però non abbiamo ancora imparato come eseguirla.

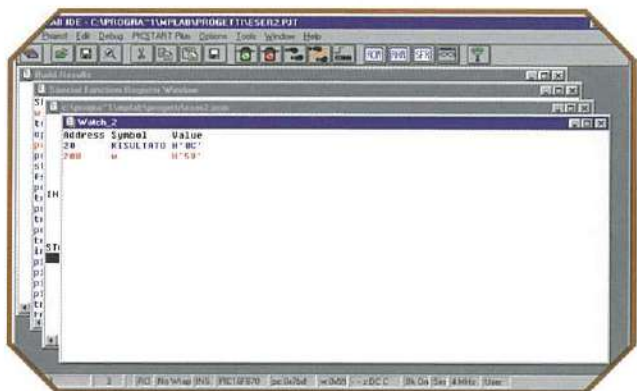
Quando attiviamo l'opzione Debug della barra del menù appare il menù a tendina mostrato nella figura della pagina successiva. Su questo menù troviamo tutte le opzioni di simulazione che analizzeremo in ordine una a una, così come abbiamo fatto con le finestre di simulazione.

## Debug → Run

Se selezioniamo questa opzione si aprirà un nuovo menù. Come possiamo vedere in questi menù a tendina, su molte delle opzioni appare a destra l'indicazione di un tasto o la combinazione di più tasti. Il software facilita l'accesso rapido alle scelte più utilizzate in modo che non sia necessario aprire i menù.



Disposizione verticale delle finestre.



Disposizione delle finestre in cascata.

Ad esempio, per selezionare l'opzione Reset, possiamo aprire questo menù o premere direttamente il tasto F6. Ricordate che è anche possibile accedere alle opzioni più utilizzate mediante le icone della barra degli strumenti.

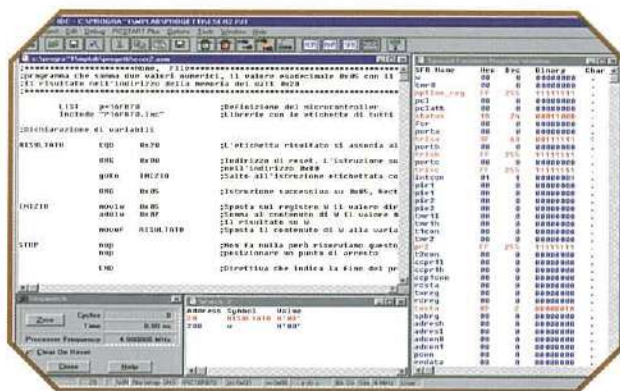
Aprirete il progetto della somma di due numeri, se non lo avete ancora aperto, assemblatelo e realizzate una disposizione sul video con le finestre di simulazione come mostrato nella figura precedente.

## Run

Attivando questa opzione si esegue il programma in tempo reale. Dato che la simulazione è fatta dal computer, l'esecuzione del programma sarà un po' più lenta rispetto all'esecuzione diretta da parte del microcontroller.



Menù dell'opzione Debug aperto.



Esempio di presentazione delle finestre sul display.

Selezionando Run vedremo cambiare il colore della barra inferiore del software e vedremo anche che il tempo indicato nella finestra Stopwatch inizierà a contare. I registri non si aggiorneranno e per fermare l'esecuzione bisognerà selezionare Halt.

## Reset

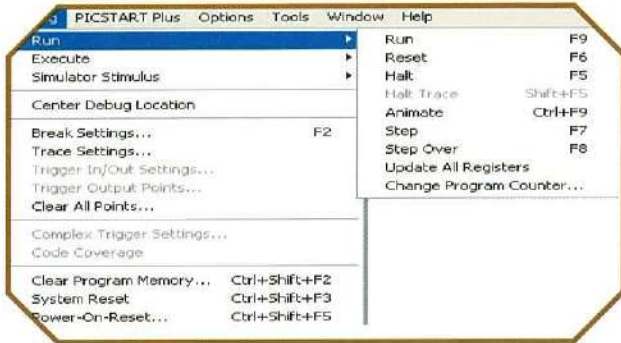
Questa opzione inizializza il sistema. Il contatore di programma (PC) viene impostato a zero e la linea di codice sorgente che corrisponde all'indirizzo del Vector di Reset risulta evidenziata. Ci posizioniamo sulla prima linea del codice di programma.

## Halt

Mediante questo comando si ferma l'esecuzione del programma. I registri si aggiornano con i valori corrispondenti all'ultima istruzione eseguita.

Applichiamo quindi le opzioni viste finora con l'esempio della somma dei due numeri. Apriamo il nostro progetto, lo assembliamo e apriamo le finestre in modo che rimanga una videata simile a una di quelle delle figure esposte in precedenza.

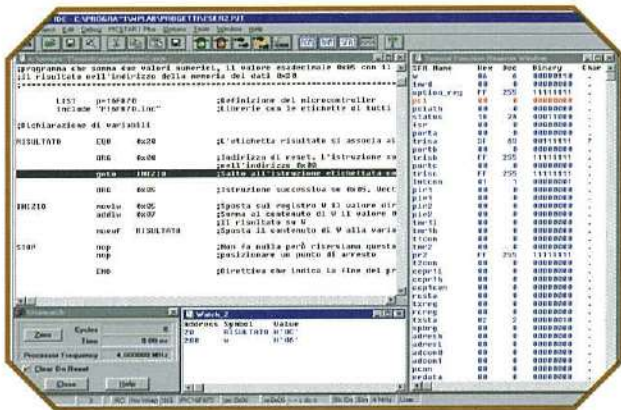
Se selezioniamo l'opzione Run vedremo la barra inferiore cambiare colore e iniziare a contare il tempo. Con Halt si ferma l'esecuzione, si ferma il tempo, cambia nuovamente il colore della barra di stato inferiore e si aggiornano i registri. La variabile risultato prende il valore finale della somma (0C in esadecimale, 12 in decimale). I registri che hanno subito modifiche cambiano colore diventando rossi. Se scegliamo l'opzione Reset il clock si impo-



Menù dell'opzione Run aperto.



Registri aggiornati dopo aver premuto Halt.



Situazione delle videate dopo un Reset.

sta a zero e cambia solamente il valore del registro Pcl che passerà a zero, indicando la posizione del Vector di Reset. La linea di codice corrispondente a questo indirizzo risulta evidenziata.

## Halt Trace

Questa opzione permette all'emulatore di caricare il buffer di memoria con i dati dei registri nei punti indicati con un Trace Point(s). Dipenderà dall'emulatore che stiamo utilizzando. Questa opzione è attiva solamente se abbiamo impostato uno o più Trace Points e non

potrà essere utilizzata se stiamo lavorando con l'emulatore.

## Animate

Il programma si eseguirà partendo dall'istruzione caricata sul PC prima di attivare questa opzione, l'esecuzione dura fino al termine del programma o fino a quando lo fermeremo con l'opzione Halt. Lavorando in questo modo la linea di stato passa a colore giallo e viene indicata l'istruzione eseguita in ogni momento.

## Step

Questa è una delle opzioni più interessanti della simulazione in quanto, grazie a essa, potremo eseguire il programma passo a passo, istruzione dopo istruzione.

Ogni volta che selezioniamo questa opzione, con la corrispondente icona della barra degli strumenti o premendo F7, MPLAB eseguirà un'istruzione del programma. Potremo così vedere come risponde ogni istruzione e se realmente esegue l'azione per cui è stata pensata. Potremo anche verificare come, dopo un Reset, selezionando Step, il valore del PC passi a 0x05, cosa che speravamo dopo l'esecuzione dell'istruzione "goto INIZIO", dove INIZIO=0x05. Rimane evidenziata l'istruzione successiva da eseguire e i valori dei registri che sono stati modificati passano al colore rosso.

## Step Over

Questa istruzione è molto simile alla precedente, dato che esegue passo a passo il programma. L'unica differenza è nel fatto che attraverso questa funzione, arrivati all'esecuzione di una subroutine (istruzione Call), questa viene eseguita come se fosse un'unica istruzione. Se la subroutine è una temporizzazione, l'istruzione equivalente impiegherà più tempo a essere eseguita rispetto un'istruzione normale. Potremo scegliere questa opzione anche mediante l'icona corrispondente o il tasto funzione F8.

Gli argomenti che trattano MPLAB e la programmazione dei microprocessori saranno inseriti sia nella sezione Digitale Avanzato che nella sezione Microprocessori



# Astabile e bistabile con porte NOR

**F**inora abbiamo fatto delle prove con le porte NAND, ora le faremo con le porte NOR e vedremo le differenze. In questo circuito si utilizzano le quattro porte dell'integrato 4001, due per formare il bistabile e le altre due per l'astabile.

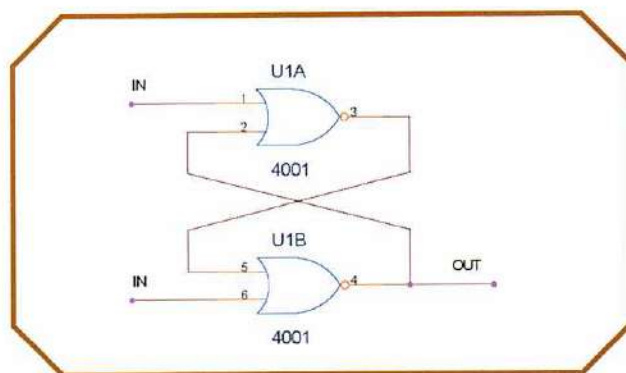
## Il circuito

Osservando lo schema possiamo vedere che le porte U1A e U1B formano il bistabile, i cui ingressi corrispondono rispettivamente ai terminali 1 e 6 dell'integrato, e che utilizziamo solamente una uscita, quella di U1B che corrisponde al terminale 4.

L'astabile è formato dalle porte U1C e U1D, l'uscita si prende dal terminale 10 e si utilizza per eccitare la base di un transistor che illumina il LED quando, sul terminale 10, vi è un livello logico alto.

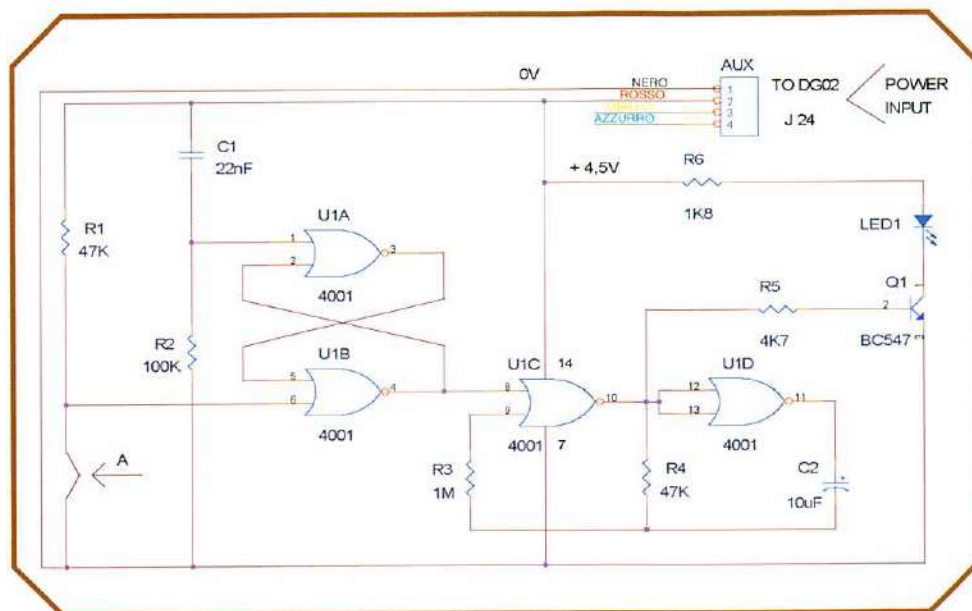
## Stato di riposo

Quando si collega l'alimentazione il collegamento A deve essere chiuso, ovvero, il terminale 6 dell'integrato deve essere allo stato di 0 logico. Se osserviamo l'altro ingresso potremo vedere che, siccome all'inizio c'è un 1, gra-

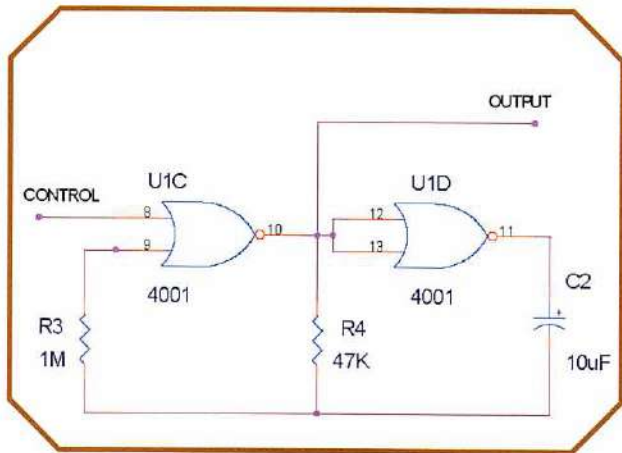


Circuito bistabile con porte NOR.

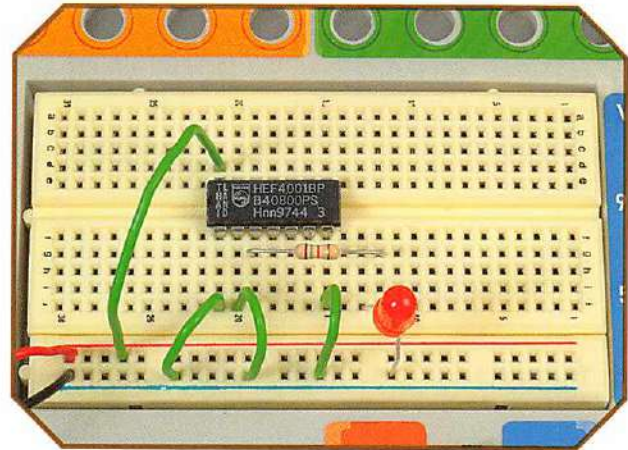
zie alla presenza del condensatore C1 collegato tra questo ingresso e il positivo dell'alimentazione (bisogna ricordare che per le porte NAND questo condensatore era collegato al negativo), l'uscita del monostabile, terminale 4, è a livello basso, l'astabile è disattivato e il LED 1 rimane spento.



Circuito astabile e bistabile con porte NOR.



Circuito astabile con porte NOR.



Montaggio per provare il 4001.

## Attivazione

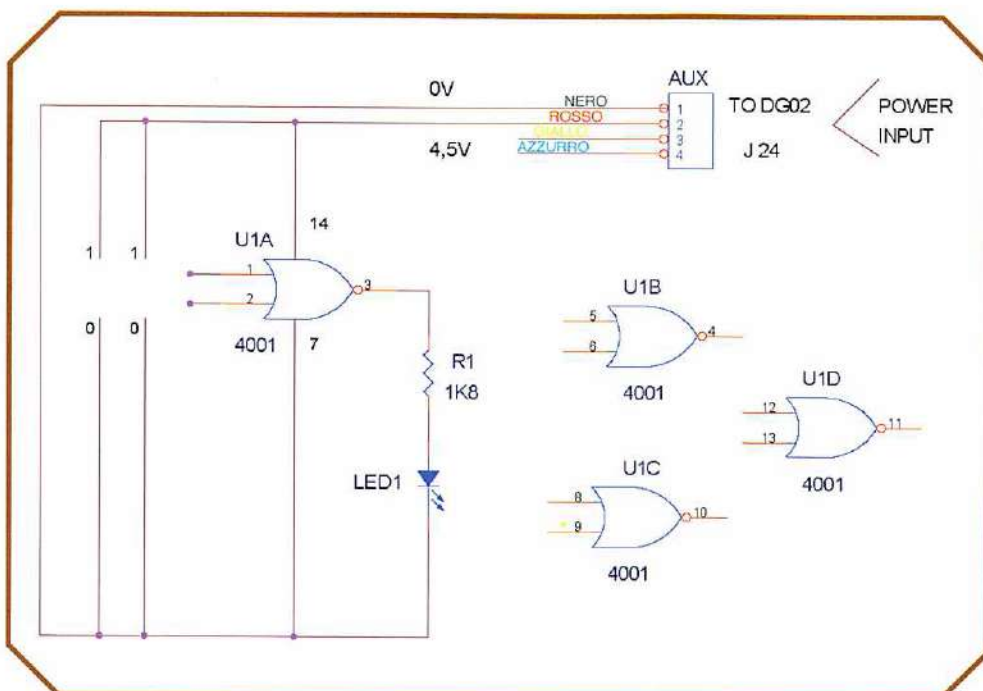
Aperto per un momento il contatto A, anche se dopo lo richiudiamo, l'uscita del bistabile, terminale 4, cambia e passa a livello alto, memorizzando questo stato fino a quando non verrà tolta l'alimentazione (in questo caso non abbiamo a disposizione il collegamento di reset). Attivando l'astabile l'illuminazione del LED sarà intermittente, con una cadenza che dipende dal valore della resistenza R4 e dalla capacità del condensatore C2.

## Prova del 4001

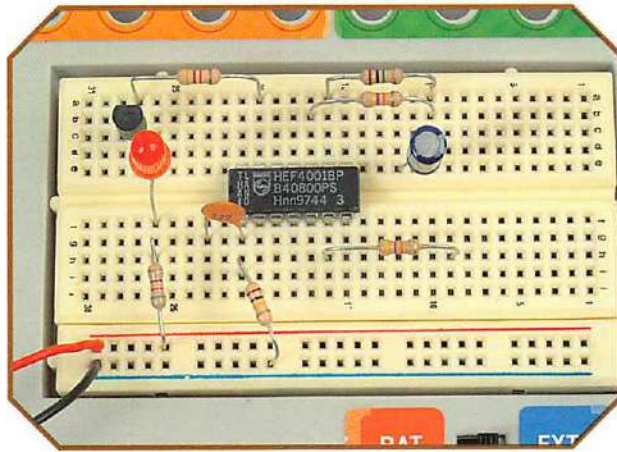
Prima di realizzare il montaggio conviene provare il circuito integrato 4001, che contiene quattro porte NOR a due ingressi. Ricordate che la sua uscita è 1 solamente quando i suoi due ingressi sono a 0, mentre per le altre due combinazioni l'uscita è 0.

L'alimentazione di questo integrato è la seguente: terminale 14 positivo e terminale 7 negativo.

Il circuito si alimenta tramite la scheda DG04 utilizzando un cavetto di collegamento



Circuito di prova per il 4001.



Componenti dell'esperimento sulla Bread Board.

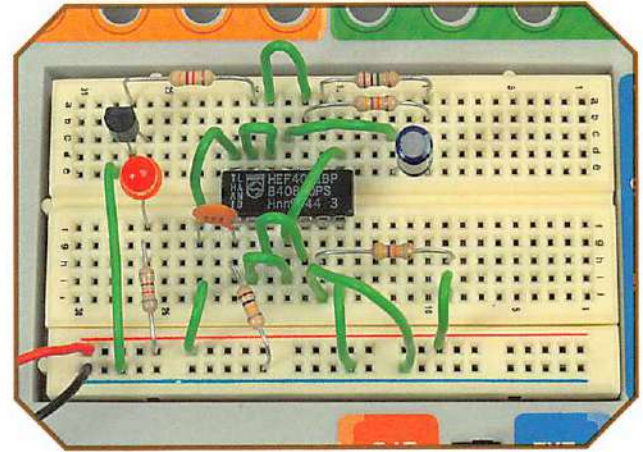
sul terminale J41, con il cavo nero al terminale 1 di questo connettore.

## Montaggio

Il montaggio del circuito del bistabile di base si realizza con la solita procedura, facendo attenzione alla polarità del LED, a quella dell'alimentazione e del condensatore elettrolitico, al montaggio del transistor e a quello del circuito integrato.

Questo montaggio è un po' più complesso e occorrerà osservare bene il codice a colori delle resistenze, per non confondere quelle da 4K7 con quelle da 47 K. Il montaggio si realizza seguendo tutti i collegamenti dello schema e aiutandosi con le fotografie.

È necessario verificare ogni filo dopo aver-

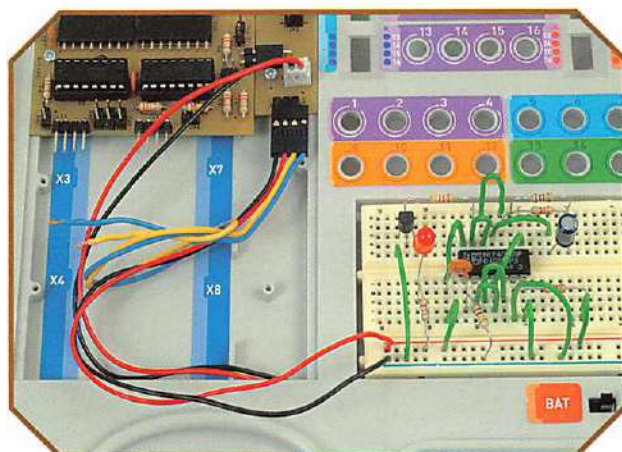


Cablaggio dell'esperimento.

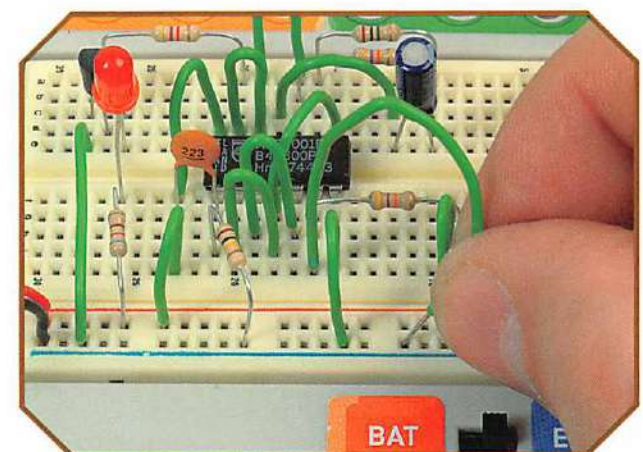
lo montato, perché è facile sbagliare, in quanto i collegamenti della Bread Board sono molto vicini fra loro; questo lavoro si può migliorare utilizzando una pinza per collegare i fili e i componenti con i terminali lunghi, come ad esempio, le resistenze e i condensatori.

## Prova

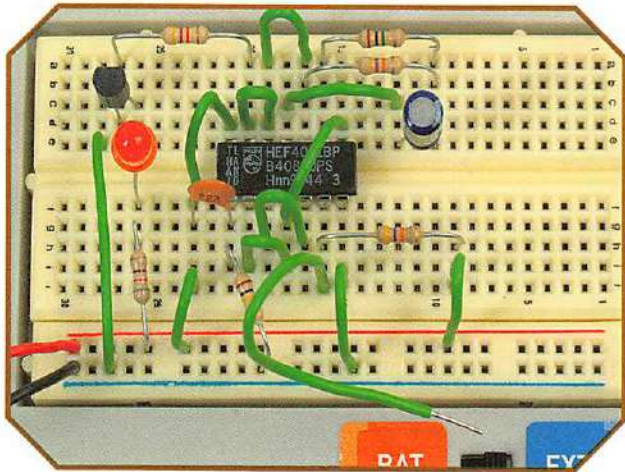
Dopo aver realizzato il montaggio e rivisto i collegamenti si lascia il collegamento A chiuso, ovvero con il filo inserito, e si collega l'alimentazione, che per il momento è fornita dai 4,5 V delle pile, anche se potrebbe funzionare senza problemi a 5 e 9 V. Il LED deve rimanere spento. Apriamo il collegamento A e lasciamolo aperto, quindi osserviamo se il



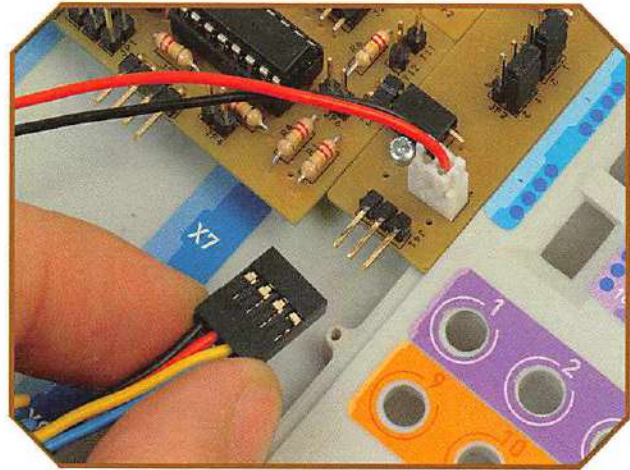
L'alimentazione si prende dalla scheda DG04.



Togliendo il collegamento si simula l'apertura del circuito.



Il LED intermittente indica lo stato di memorizzazione del bistabile.



Il reset si ottiene scollegando l'alimentazione.

LED si illumina in modo intermittente. Collegando nuovamente A vedremo che il LED si mantiene intermittente, quindi abbiamo verificato che il bistabile ha memorizzato questo stato di allarme. Lo stato di allarme si cancella scollegando l'alimentazione del circuito e ricollegandola successivamente, durante questa operazione il collegamento A deve essere chiuso.

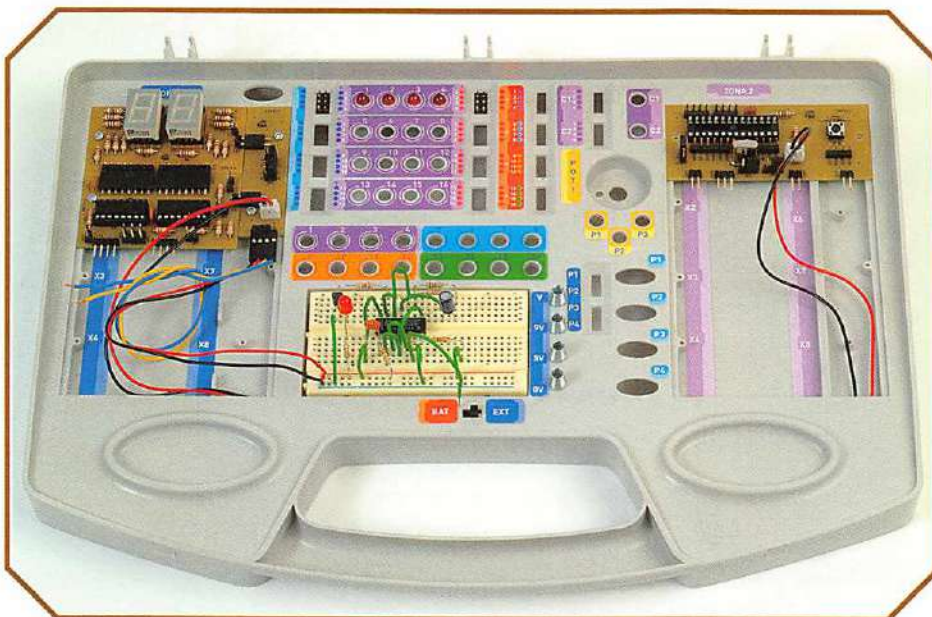
#### LISTA DEI COMPONENTI

Circuito di base	
U1	Circuito integrato 4001
R1,R4	Resistenza 47 K (giallo, viola, arancio,
R2	Resistenza 100 K (marrone, nero, giallo)
R3	Resistenza 1 M (marrone, nero, verde)
R5	Resistenza 4K7 (giallo, viola, rosso)
R6	Resistenza 1K8 (marrone, grigio, rosso)
C1	Condensatore 22 nF
C2	Condensatore 10 $\mu$ F elettrolitico
LED1,LED2	Diodo LED rosso

## Modifiche

In questo circuito si può e si deve realizzare qualche esperimento eseguendo alcune modifiche. La cadenza dell'intermittenza dipende dai valori di C2 e R4, quindi se montiamo, ad esempio, due condensatori da 10  $\mu$ F in serie, la frequenza raddoppia.

È anche possibile aggiungere un collegamento di reset, mantenendo chiuso il collegamento A e scaricando il condensatore C1 collegando momentaneamente un filo ai suoi capi; in questo modo applichiamo un 1 al terminale 1 dell'integrato.



Vista generale dell'esperimento.





## Bistabile tipo T

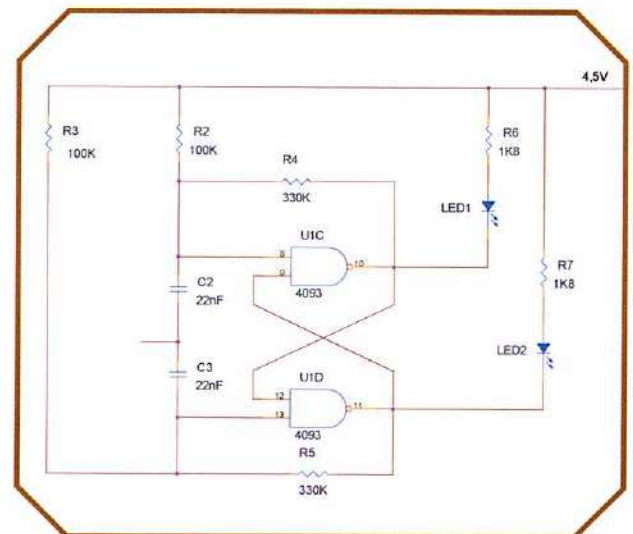
**U**tilizzeremo nuovamente il circuito integrato 4093 con le sue quattro porte NAND a due ingressi, per realizzare un circuito di prova.

### Il circuito

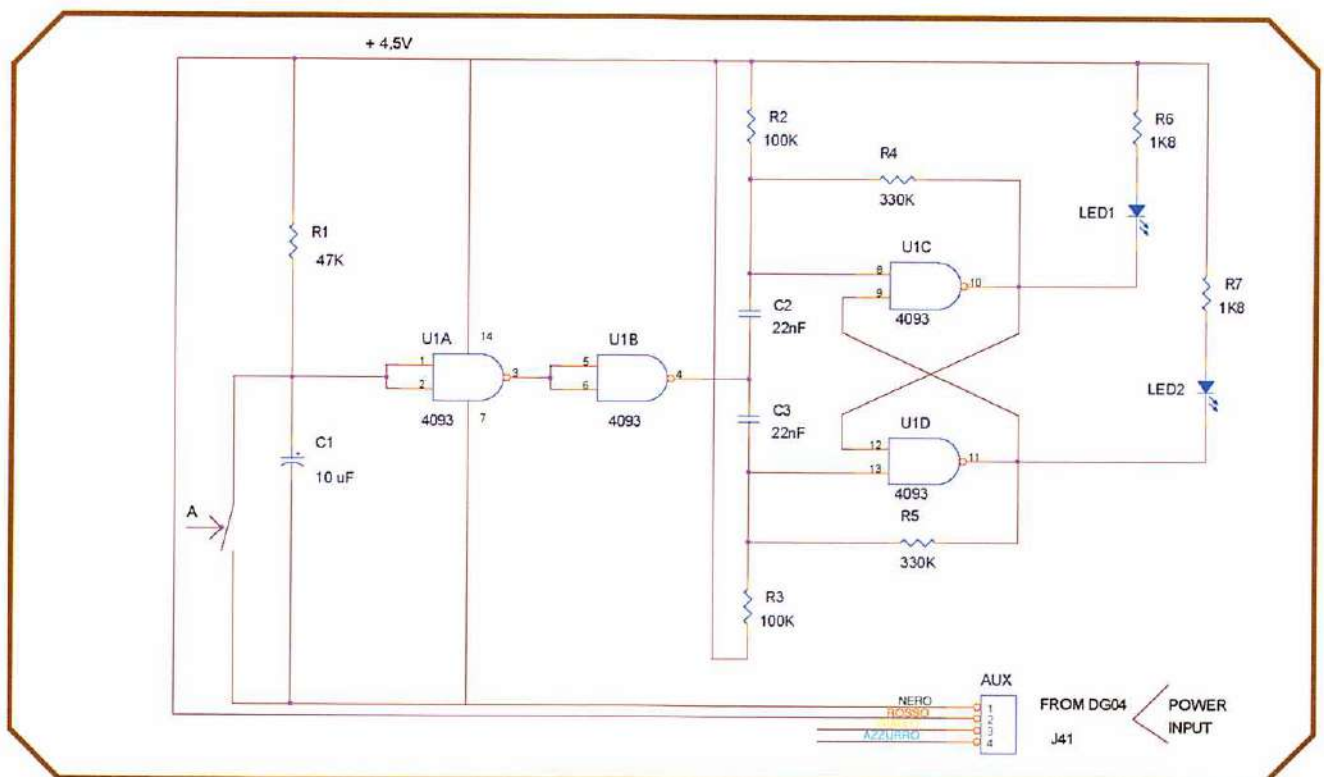
Osservando lo schema della figura in basso, da sinistra verso destra, vediamo un interruttore ai capi di un condensatore, il cui terminale positivo è collegato al positivo dell'alimentazione tramite la resistenza R1. L'interruttore, che al momento giusto potrà essere sostituito da un pulsante, si utilizza per scaricare il condensatore, e quando si libera, il condensatore si carica nuovamente.

Questo condensatore è collegato all'ingresso di U1A, che si configura come porta invertente, avendo entrambi gli ingressi collegati insieme.

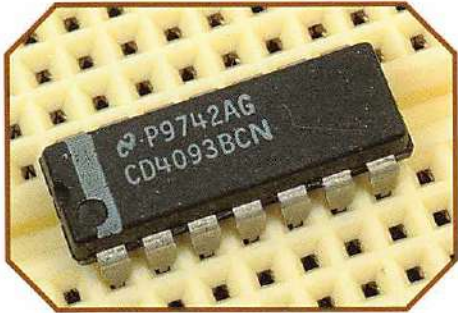
Terminata la fase transitoria successiva al collegamento dell'alimentazione il circuito si trova in stato di riposo completamente stabi-



I LED di uscita si illuminano quando l'uscita è a zero.



Circuito bistabile tipo T.



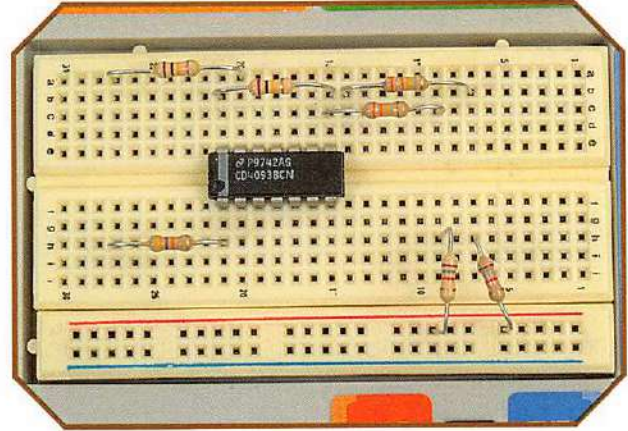
Montaggio del circuito integrato.

lizzato, il condensatore si è caricato e sui terminali 1 e 2 troviamo una tensione uguale a quella dell'alimentazione, ovvero un livello logico alto; l'uscita di questa porta invertente, terminale 3 del circuito integrato, è quindi a livello basso.

Questa uscita si collega direttamente ai terminali 5 e 6 dell'integrato, che sono uniti tra loro e costituiscono l'ingresso della porta U1B, anch'essa con un funzionamento invertente; quindi, nello stato attuale del circuito, dato che sul suo ingresso c'è un livello logico basso, sulla sua uscita troviamo un livello logico alto.

Le porte U1C e U1D formano un classico bistabile; in questo tipo di circuito una delle uscite è 0 e l'altra 1, dato che una è sempre l'inverso, o la negazione, dell'altra.

Trattandosi di un circuito simmetrico, le piccole differenze tra i valori dei componenti fanno in modo che allo stato iniziale si illumini uno o l'altro LED, corrispondente all'una o all'altra uscita.

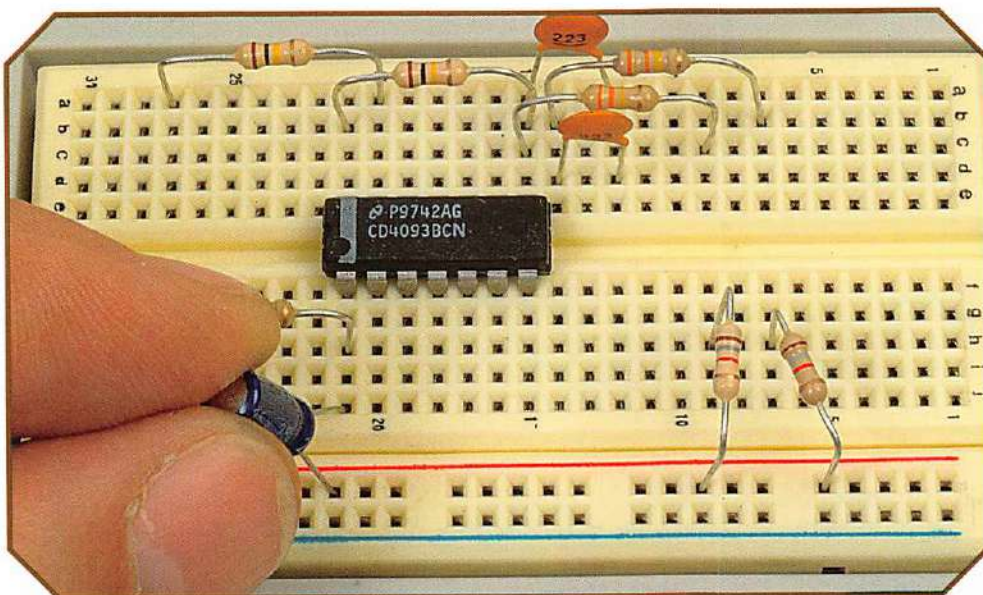


Scheda Bread Board con le resistenze.

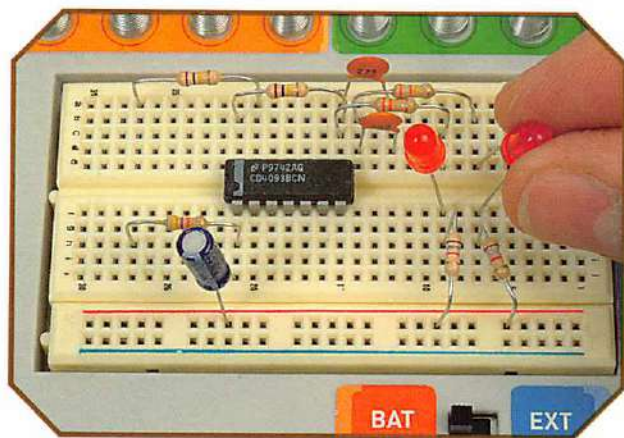
Le resistenze R2 e R3 mantengono un livello alto sugli ingressi per fare in modo che ogni volta che si riceve un impulso di clock l'uscita cambi, ovvero, se è a livello alto passa a livello basso e viceversa. I condensatori si utilizzano per trasmettere l'impulso che genera il cambio.

## Attivazione

Premendo per un istante il pulsante A, anche se per ora non abbiamo a disposizione il pulsante e il collegamento deve essere realizzato manualmente con un pezzo di filo, si applica per un attimo un livello basso all'ingresso di U1A, facendo passare la sua uscita a livello alto, e l'uscita di U1B a livello basso, dopodiché nuovamente a livello alto; queste transizioni di nivel-



Montaggio dei tre condensatori dell'esperimento.



Inserendo i LED bisogna rispettarne la polarità.

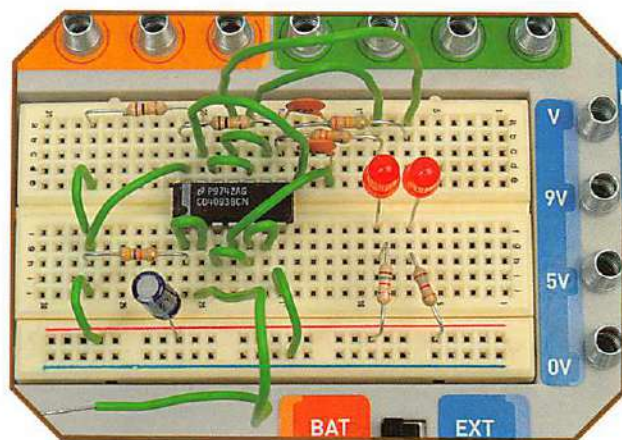
lo si propagano tramite i condensatori C2 e C3 agli ingressi del bistabile, terminali 8 e 13 del circuito integrato, e fanno in modo che questo cambi invertendo lo stato delle uscite, quindi il LED che era illuminato si spegne e quello che era spento si illumina, rimanendo in questo stato fino a quando non realizzeremo nuovamente il collegamento su A.

## Memoria

Il bistabile tipo T è una cella elementare di memoria che, come abbiamo visto, memorizza l'impulso: a ogni impulso di A, l'uscita cambia stato e rimane così fino al prossimo impulso o fino a quando si mantiene l'alimentazione.

## Montaggio

Il montaggio del circuito del bistabile di base non è difficile, però deve essere realizzato se-



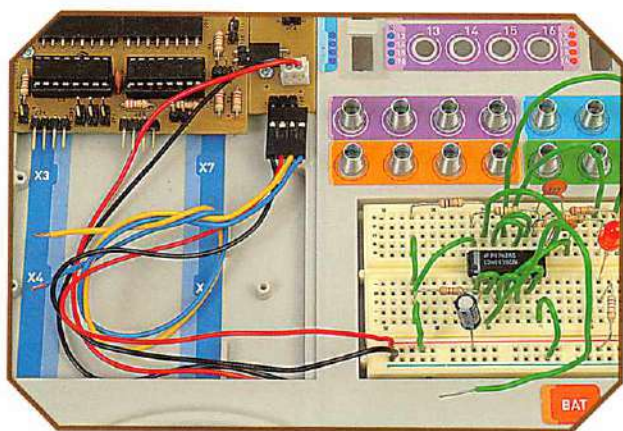
Cablaggio della Bread Board.

guendo un certo ordine, per evitare di commettere errori.

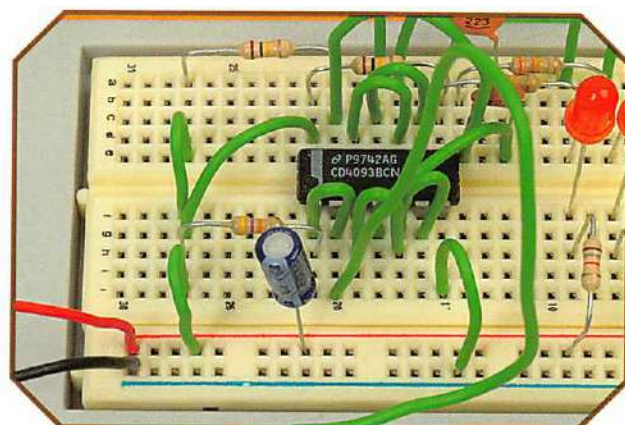
Vi consigliamo di seguire lo schema e sfruttare il grande aiuto offerto dalle fotografie, dato che un'adeguata distribuzione e collocazione dei componenti facilita il montaggio dell'esperimento; acquisendo una certa dimestichezza questo lavoro verrà svolto in modo molto rapido. Dobbiamo sempre ricordare i collegamenti interni della scheda Bread Board, ricordandone i principali: le cinque connessioni di ogni colonna sono unite tra loro.

Nella lista dei componenti è citato anche il colore degli anelli che indicano il valore della resistenza, al fine di evitare errori. L'ultimo anello corrisponde alla tolleranza, è di colore oro, per il 5%, anche se non è menzionato nella lista.

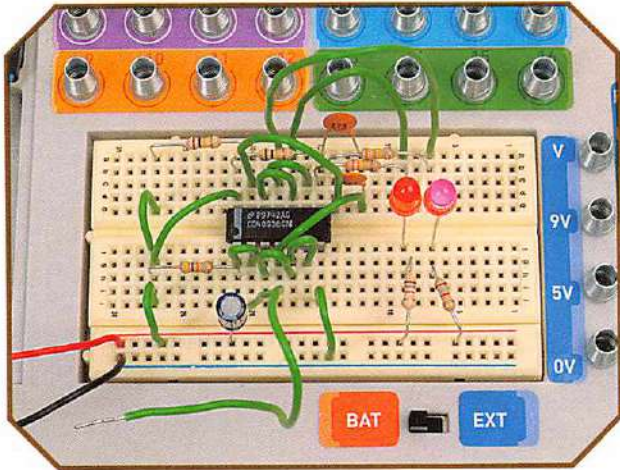
Bisogna fare attenzione all'orientamento del circuito integrato e alla polarità dei LED, dell'alimentazione e del condensatore elettrolitico.



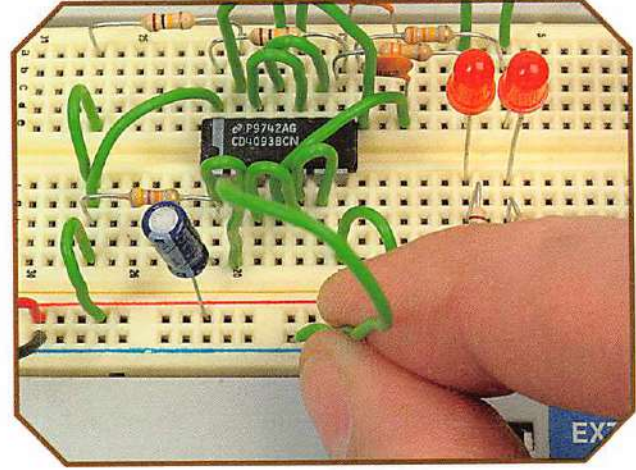
L'alimentazione si prende dalla scheda DG04.



Polo rosso al positivo e nero al negativo.



I LED indicano lo stato delle uscite.



Il collegamento A si realizza toccando per un attimo con questo filo.

Dopo aver terminato il montaggio e prima di collegare l'alimentazione è necessario verificare tutto il lavoro svolto, in quanto, anche se si tratta di un circuito piccolo, ha un

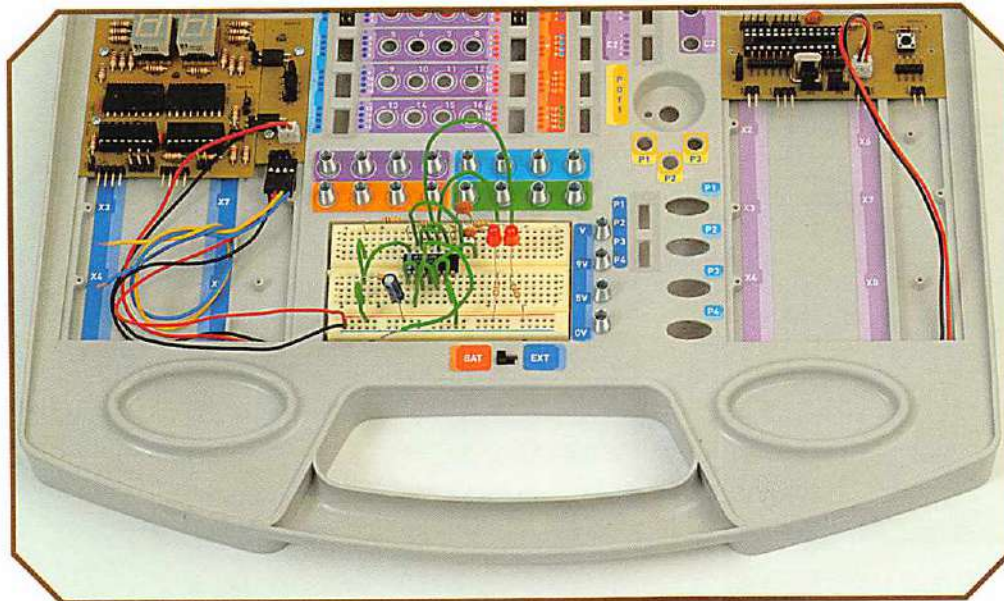
#### LISTA DEI COMPONENTI

Circuito di base	
U1	Circuito integrato 4093
R1	Resistenza 47 K (giallo, viola, arancio)
R2, R3	Resistenza 100 K (marrone, nero, giallo)
R4, R5	Resistenza 1 M (marrone, nero, verde)
R6, R7	Resistenza 1K8 (marrone, grigio, rosso)
C1	Condensatore 10 $\mu$ F elettrolitico
C2, C3	Condensatore 22 nF
LED1, LED2	Diodo LED rosso

considerevole numero di punti di collegamento ed è relativamente facile commettere qualche piccolo errore di collegamento, dimenticare qualche componente o semplicemente omettere qualche connessione.

#### Prova

Dopo aver verificato il lavoro si può collegare l'alimentazione. Uno dei LED si deve illuminare, infatti eseguendo il collegamento A si cambiano i LED, ovvero, quello che era spento si illumina e quello che era illuminato si spegne. Questo cambio si ripete ogni volta che si realizza il collegamento A.



Vista generale dell'esperimento.



# Bistabile tipo D

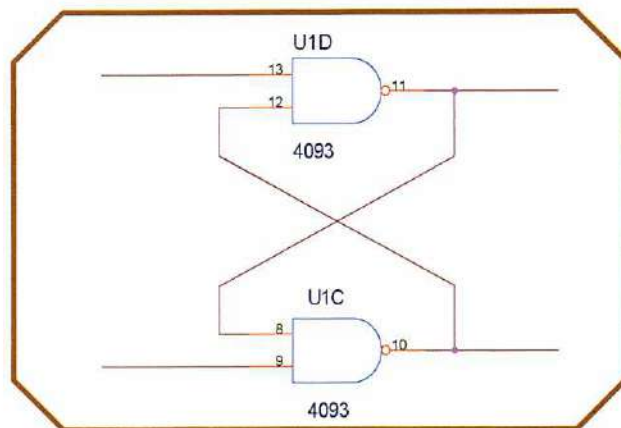
**Q**uesto esperimento consiste nel costruire un bistabile tipo D sincrono. Questo tipo di bistabile ha un ingresso per i dati e l'altro per il clock. Il dato presente sull'ingresso può essere solamente 0 o 1, e si trasferisce all'uscita unicamente quando si applica un impulso di clock. Se non c'è l'impulso di clock non c'è trasferimento del dato. Quando cessa l'impulso, lo stato dell'uscita si mantiene.

## Il circuito

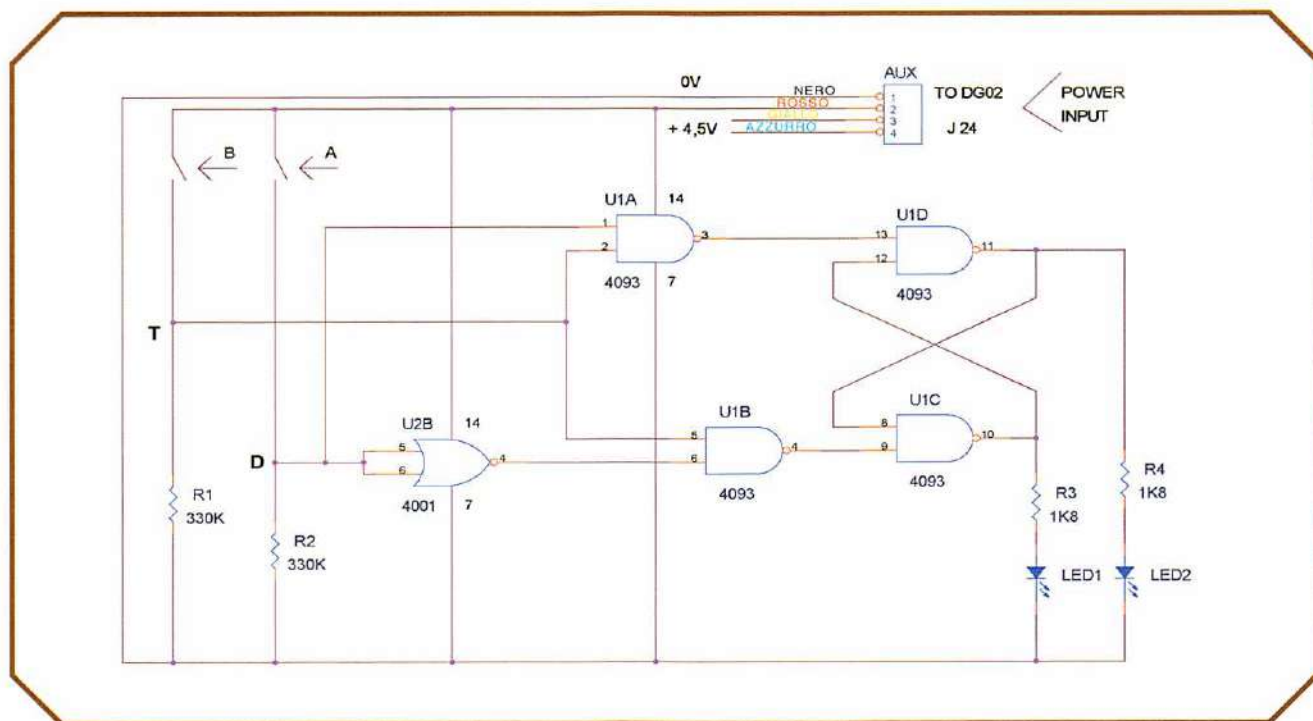
Se osserviamo attentamente lo schema in basso vedremo che il circuito di base parte da un bistabile RS, formato dalle porte U1D e U1C; le loro uscite sono rispettivamente i terminali 11 e 10 del circuito integrato 4093.

## Bistabile RS sincrono

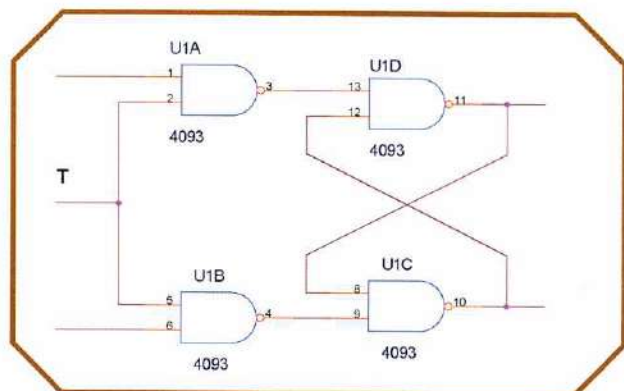
Le altre due porte di questo integrato, ovvero U1A e U1B, le cui uscite si applicano a entrambi gli ingressi del bistabile, sono utilizzate come elementi di sincronizzazione, infatti possiamo notare che un ingresso di ogni porta, il



Bistabile RS di base.



Circuito bistabile tipo D.



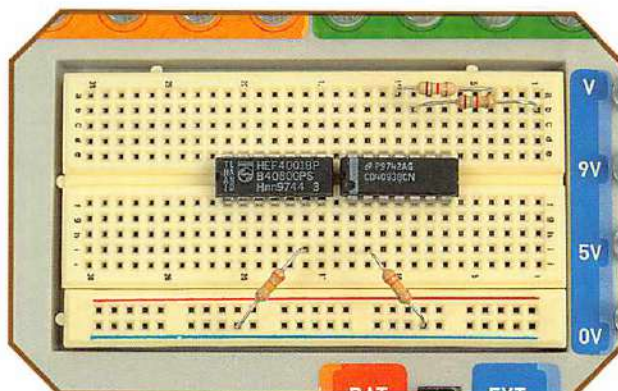
Bistabile RS con ingresso di sincronizzazione (T).

2 di U1A e il 5 di U1B, è unito all'ingresso T di sincronizzazione, che si può anche chiamare ingresso di clock. In questo modo si ottiene un bistabile RS sincrono.

### Bistabile D sincrono

La porta NOR U2B è collegata tra i due ingressi RS, in modo che uno sia sempre invertito rispetto all'altro. Bisogna tener presente che un bistabile tipo D ha solamente un ingresso, e lo otteniamo in questo modo utilizzando un bistabile RS.

Quando i due collegamenti A e B sono aperti, le resistenze R1 e R2 mantengono lo stato



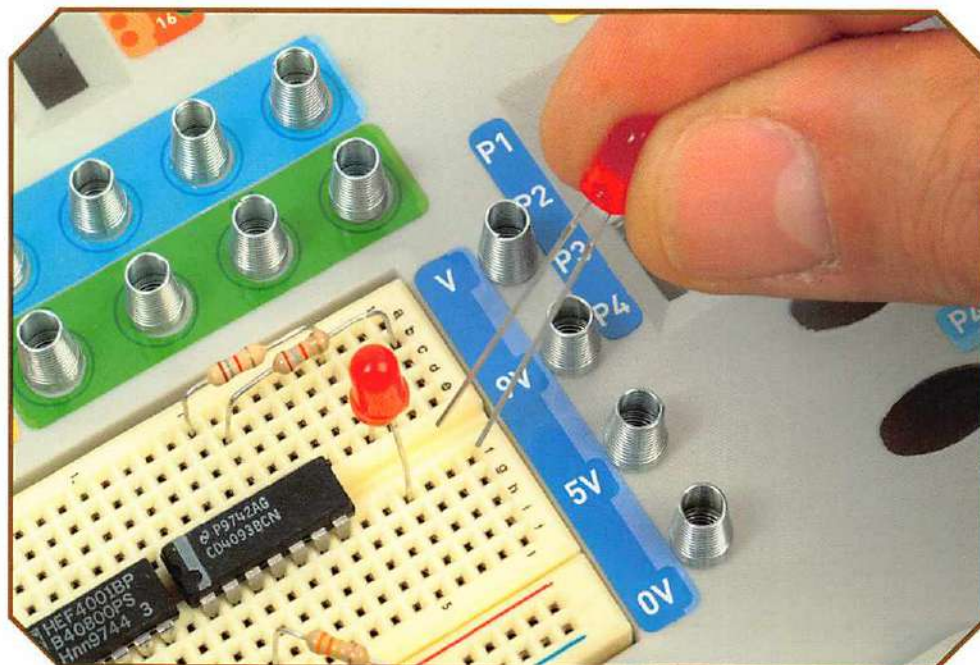
Componenti montati sulla scheda Bread Board.

a 0. Chiudendo il collegamento si applica un 1 all'ingresso corrispondente.

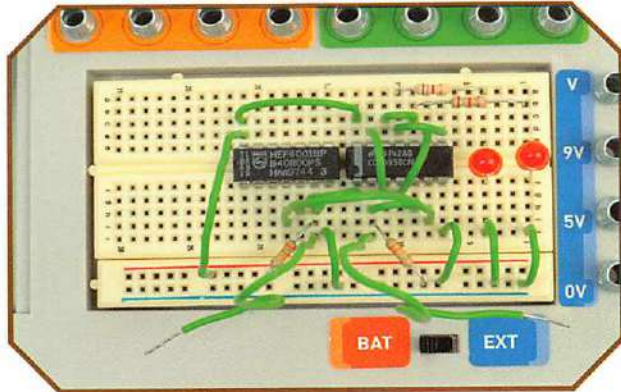
Infine i due LED, con le loro resistenze di limitazione per la corrente, si utilizzano come indicatori dello stato di ogni uscita, utilizzando le due uscite del bistabile illumineranno solamente un LED alla volta.

### Montaggio

Il montaggio di questo esperimento è quello abituale, i primi componenti da inserire sono i due circuiti integrati, dopo monteremo il resto dei componenti e realizzeremo i collegamenti, senza dimenticare le alimentazioni



Bisogna fare attenzione alla polarità dei LED: l'anodo è il terminale più lungo.

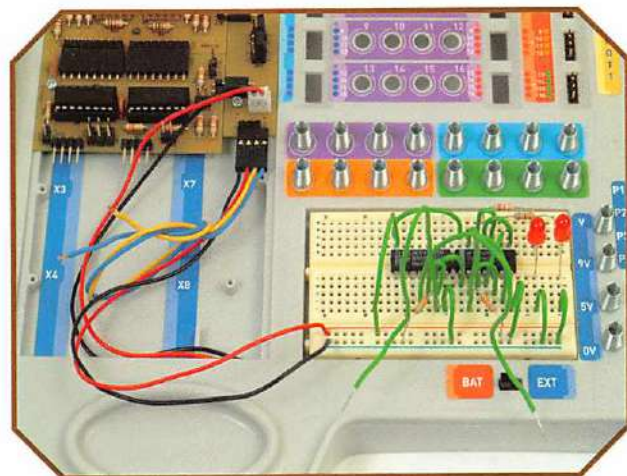


Esperimento con il cablaggio realizzato.

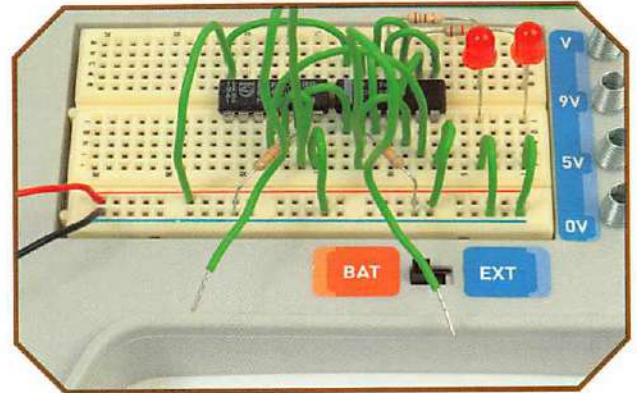
dei due integrati, terminali 14 e 7 rispettivamente, e facendo attenzione alla polarità dei LED. L'alimentazione si ottiene dal connettore J24 di DG02. Quando il laboratorio sarà più completo e le molle 0 e 9 saranno collegate, da queste ultime potremo prendere l'alimentazione.

Terminato il montaggio, prima di collegare l'alimentazione, conviene rivedere tutto il lavoro svolto, verificando che ogni integrato sia quello indicato, che i LED siano collegati con la polarità adeguata e che tutti i collegamenti siano corretti. Tenete presente che è piuttosto facile sbagliare, dato che si lavora in una zona molto piccola con collegamenti molto vicini.

Bisogna anche fare attenzione a evitare cortocircuiti tra i terminali dei componenti non isolati e lunghi, come quelli delle resistenze.



Alimentazione utilizzata.



I collegamenti A e B si fanno con fili.

L'alimentazione verrà collegata dopo aver rivisto tutto il lavoro di montaggio svolto.

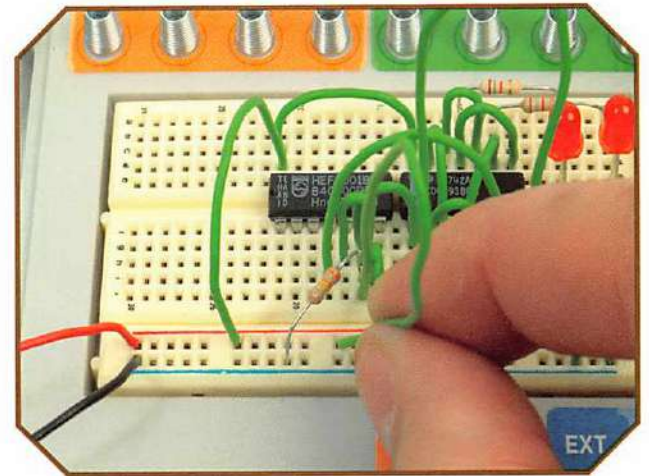
## Prova di funzionamento

Dopo aver verificato il circuito collegheremo l'alimentazione, e a questo punto, uno dei due LED si illuminerà e l'altro rimarrà spento.

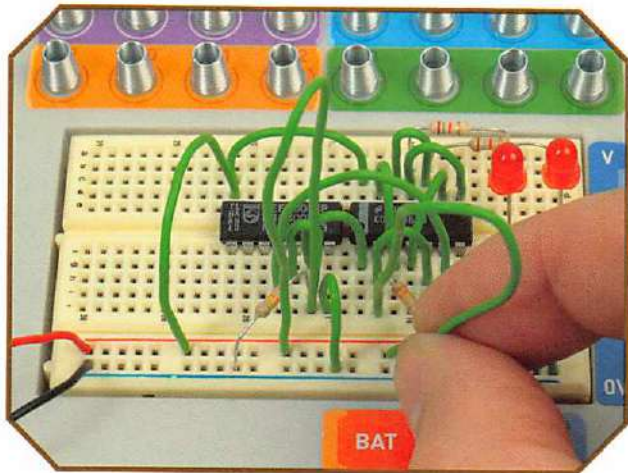
Partiamo dalla posizione in cui i collegamenti A e B sono aperti, ovvero i fili che utilizziamo per rappresentarli sono scollegati. Colleghiamo e scollegiamo A diverse volte e vedremo che lo stato del LED non cambia.

Ora colleghiamo B e osserviamo che ogni volta in cui si collega e si scollega, l'ingresso si trasferisce all'uscita.

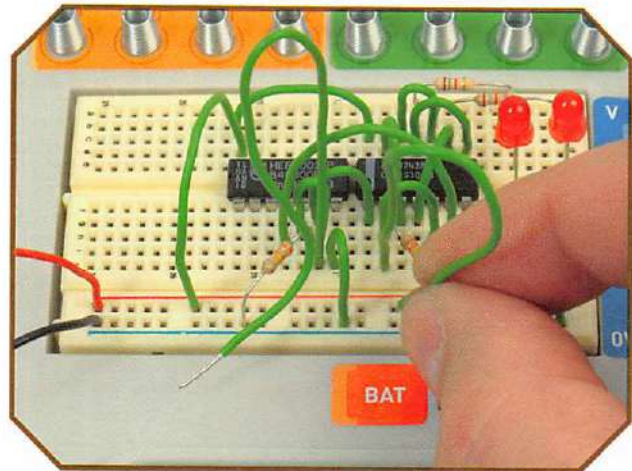
Con B chiuso cambiamo lo stato di A e osser-



Collegando A si applica un 1 all'ingresso del bistabile.



Collegando B  
l'ingresso si trasferisce all'uscita.



Scollegando A e collegando nuovamente B, anche se per breve tempo, l'uscita cambia.

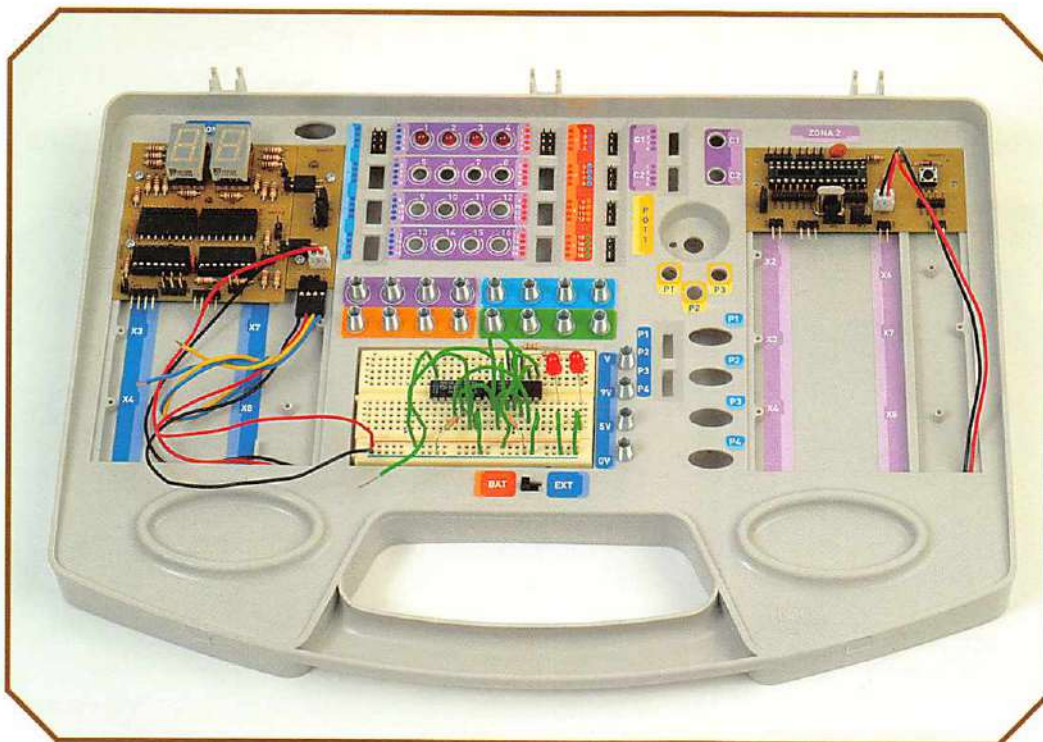
#### LISTA DEI COMPONENTI

Circuito di base	
U1	Circuito integrato 4093
U2	Circuito integrato 4001
R1, R2	Resistenza 330 K (arancio, arancio, giallo)
R3, R4	Resistenza 1K8 (marrone, grigio, rosso)
LED1, LED2	Diodo LED rosso

viamo ciò che succede. Quando l'uscita è cambiata rispetto allo stato iniziale, stacciamo B e vediamo che questa nuova situazione si memorizza.

Con B aperto, ovvero scollegato, colleghiamo diverse volte A e vedremo che l'uscita non cambia, perché, essendo aperto B, l'ingresso non si trasferisce all'uscita.

Riassumendo, bisogna collegare B per fare in modo che l'ingresso si trasferisca all'uscita.



Vista generale del  
laboratorio  
con l'esperimento.





## Decodificatore da 2 a 4

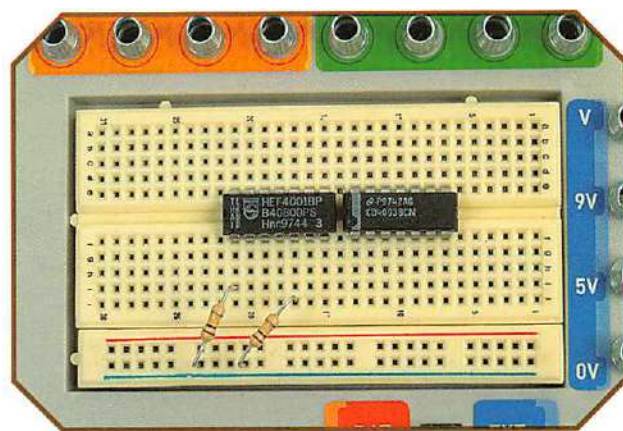
**Q**uesto esperimento consiste nell'utilizzare le quattro combinazioni che si possono formare con due bit per selezionare una delle quattro uscite.

Ogni bit di ingresso si rappresenta con un collegamento, in totale due collegamenti, indicati come A e B, che rappresentano lo stato logico uno a collegamento eseguito, e lo stato logico zero quando sono scollegati.

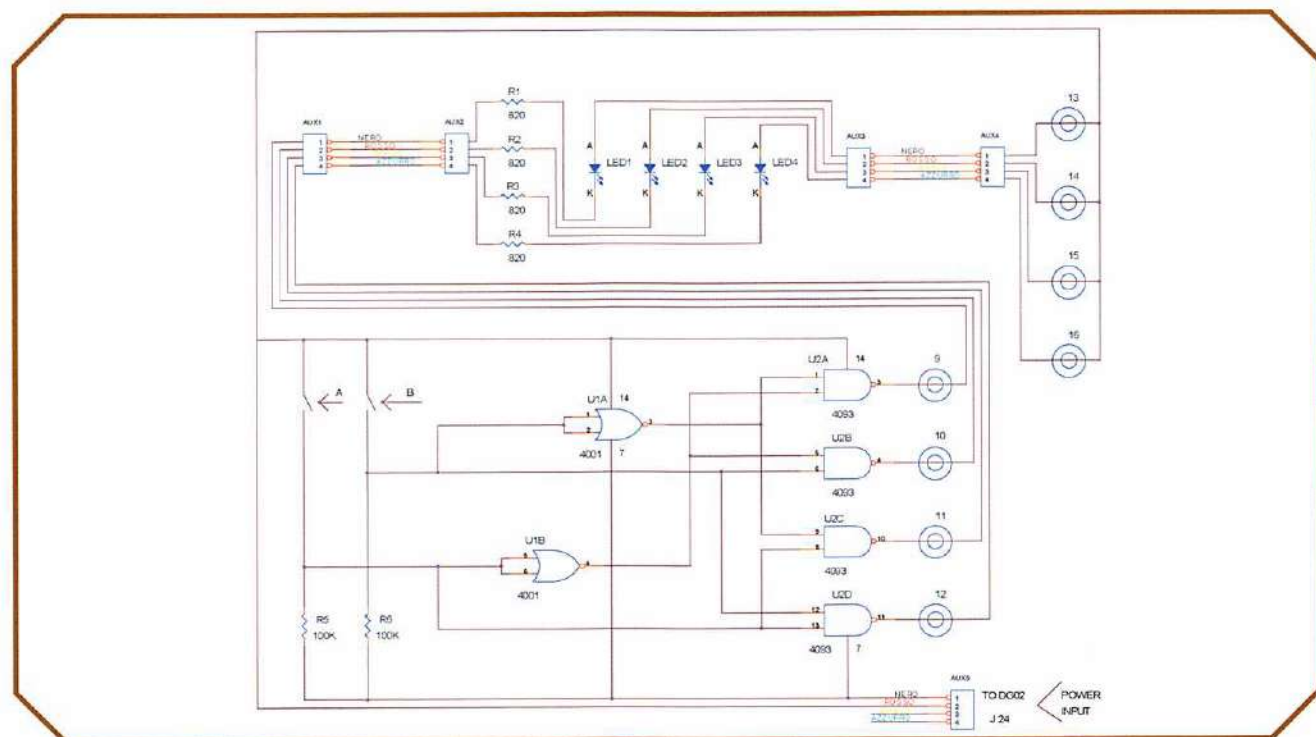
### Il circuito

Per capire il funzionamento del circuito faremo riferimento allo schema. A prima vista potrebbe sembrare complicato, però osservando attentamente potremo identificare le parti di cui è composto.

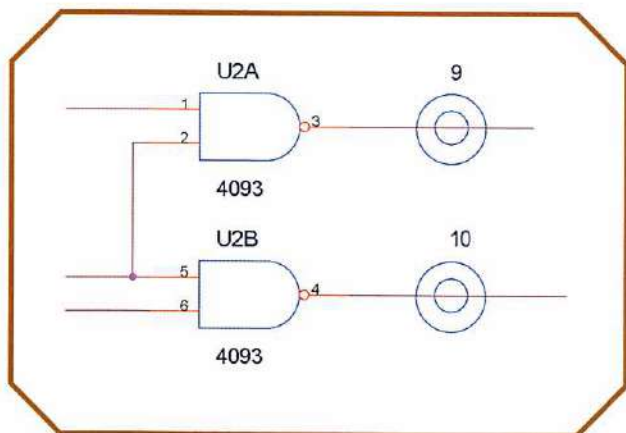
Nella parte bassa dello schema possiamo vedere il cavetto di collegamento che si connette al terminale J24 della scheda DG02; il filo nero si utilizza per il negativo dell'alimentazione e si collega ai terminali 7 di entrambi gli integrati, mentre il positivo, ovvero il filo rosso, è l'alimentazione da 5 V (4,5 V dato che utilizziamo le tre pile).



Componenti montati sulla scheda Bread Board.



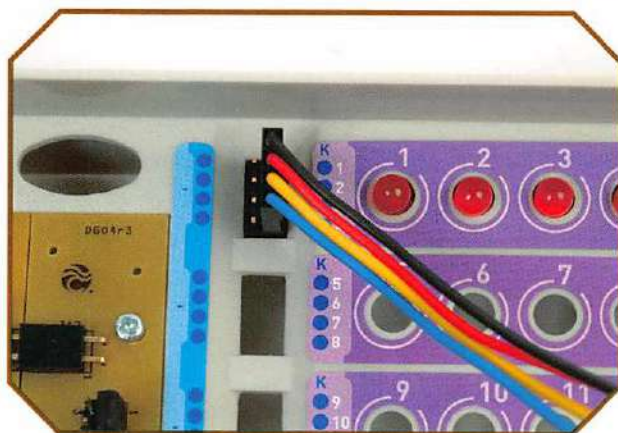
Decodificatore da 2 a 4.



Le uscite si collegano direttamente alle molle, che sono raffigurate mediante due circonferenze e il loro numero corrispondente.

Osservando attentamente vedremo che l'alimentazione da 5 volt, oltre ad alimentare i circuiti integrati, continua verso la parte superiore dello schema e arriva contemporaneamente ai terminali delle molle siglati 13, 14, 15 e 16; questi collegamenti passano nella parte inferiore del pannello frontale del laboratorio e vanno ai terminali del connettore della scheda DG12, anch'essi siglati come 13, 14, 15 e 16. Da questo connettore, con un cavetto terminato in due connettori, si arriva agli anodi dei LED 1, 2, 3 e 4, sul connettore siglato come A 1, 2, 3 e 4.

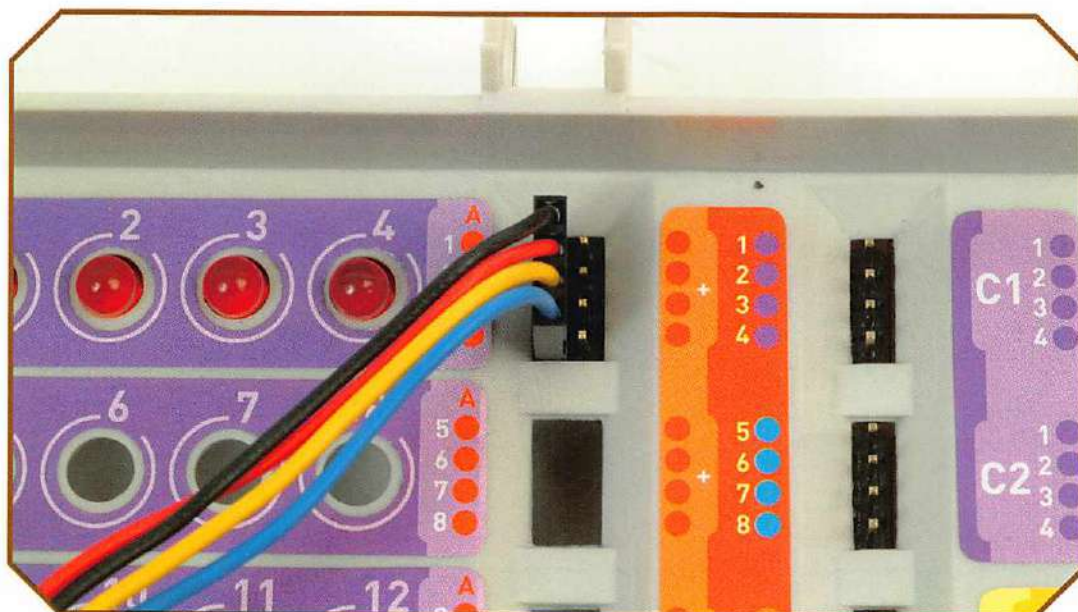
I quattro LED rappresentati nello schema,



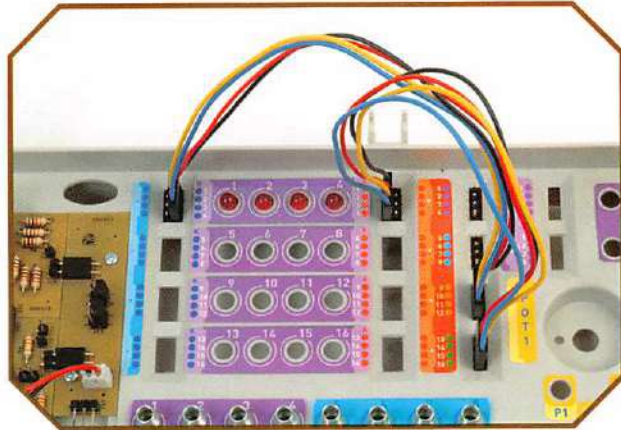
Dettaglio ingrandito del collegamento tra i catodi dei LED e il cavetto.

così come le quattro resistenze da R1 a R4, da 820  $\Omega$ , fanno parte della scheda DG11, quindi non sono riportati sulla lista dei componenti, dato che su questa lista vengono indicati solamente i componenti montati sulla scheda Bread Board.

Le resistenze di limitazione dei LED sono collegate al connettore siglato come K 1, 2, 3, 4, da questo connettore giungono ai terminali 9, 10, 11 e 12 della scheda DG12, con un cavetto terminato su due connettori a quattro vie ognuno, si collegano ai terminali 9, 10, 11 e 12 che sono internamente collegati alle molle 9, 10, 11 e 12.



Dettaglio del collegamento degli anodi.



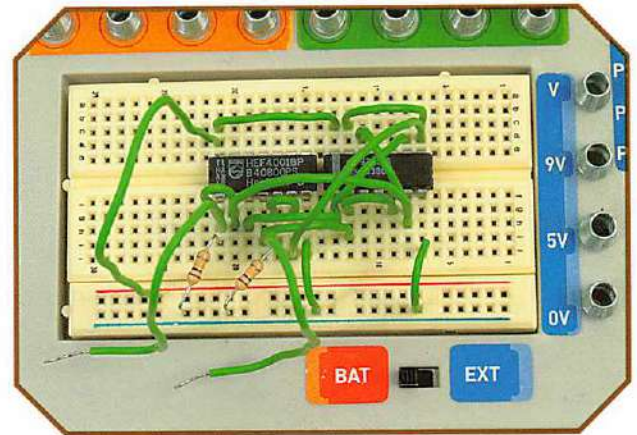
I collegamenti tra i terminali da 1 a 16 e le molle sono permanenti e si realizzano nella parte inferiore del pannello principale del laboratorio.

### Ingressi

Gli ingressi al circuito sono rappresentati dai collegamenti A e B. Le resistenze R5 e R6 mantengono l'ingresso del circuito a 0 quando il collegamento è aperto. Quando si realizza il collegamento A o B, si applica un 1 all'ingresso del circuito, in modo che le resistenze R5 e R6 rimangano collegate tra il positivo e il negativo dell'alimentazione consumando poca corrente.

Le quattro combinazioni possibili sono:

A	B	LED illuminato
0	0	LED 1
0	1	LED 2
1	0	LED 3
1	1	LED 4



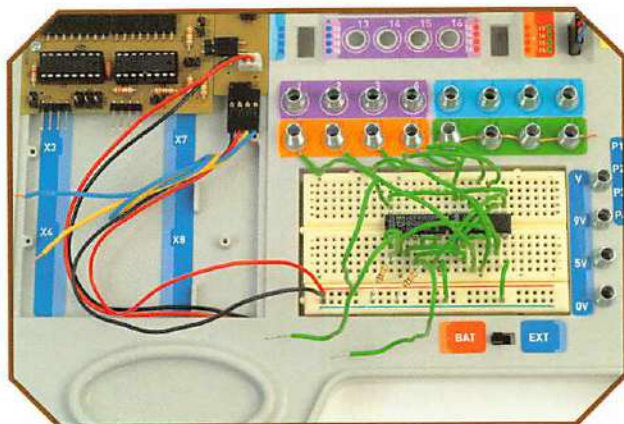
Esperimento con il cablaggio realizzato. I collegamenti A e B sono anch'essi eseguiti con fili.

### Montaggio

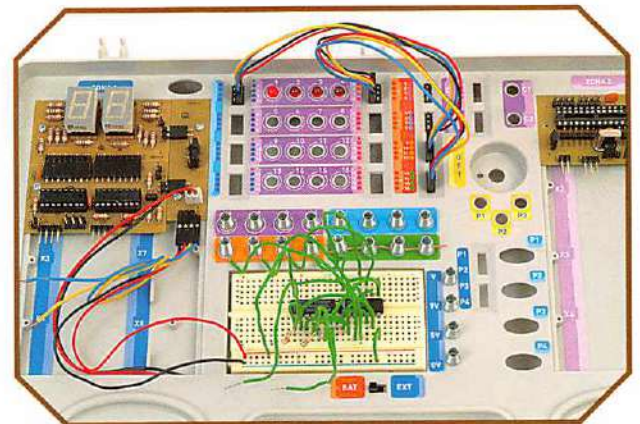
Il montaggio di questo esperimento presenta alcune innovazioni dato che è già montato il sistema di collegamento ed è anche possibile utilizzare le molle di interconnessione.

Si utilizzano i quattro LED del pannello frontale e i due cavetti di collegamento, come vi è stato indicato, per eseguire le connessioni di questi LED alle molle di interconnessione. Il montaggio dei componenti sulla scheda Bread Board si realizza come da abitudine, seguendo lo schema e aiutandoci con le fotografie.

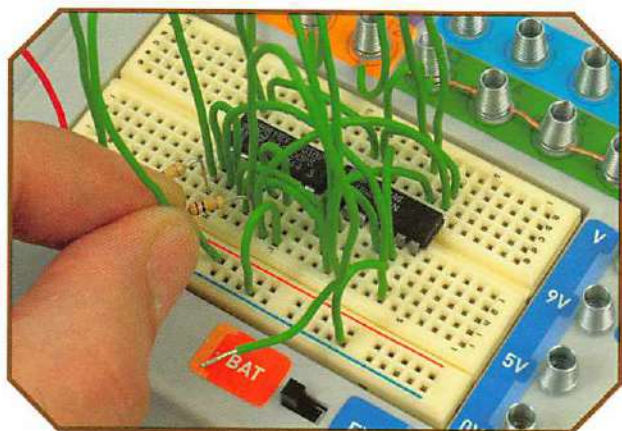
I collegamenti tra questa scheda e le molle sono realizzati direttamente con un filo spezzato su entrambi i lati, è sufficiente inclinare



Cablaggio delle molle e alimentazione.



Con A e B scollegati si accende il LED 1.



Collegando B, con A scollegato, si illumina il LED 2.

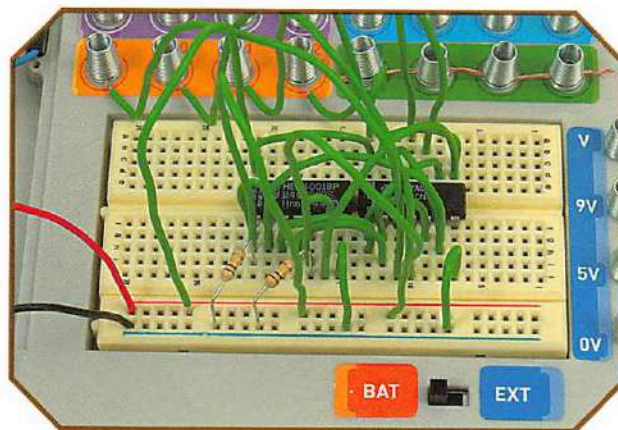
leggermente la molla per far separare le sue spire, inserire l'estremo del filo e rilasciare la molla per fissarlo.

È necessario utilizzare molti fili quindi, per facilitare i collegamenti e verificare la versatilità delle molle nelle connessioni, è stato utilizzato un avanzo di filo senza isolante per unire fra di loro le molle dalla 13 alla 16.

#### LISTA DEI COMPONENTI

Circuito di base

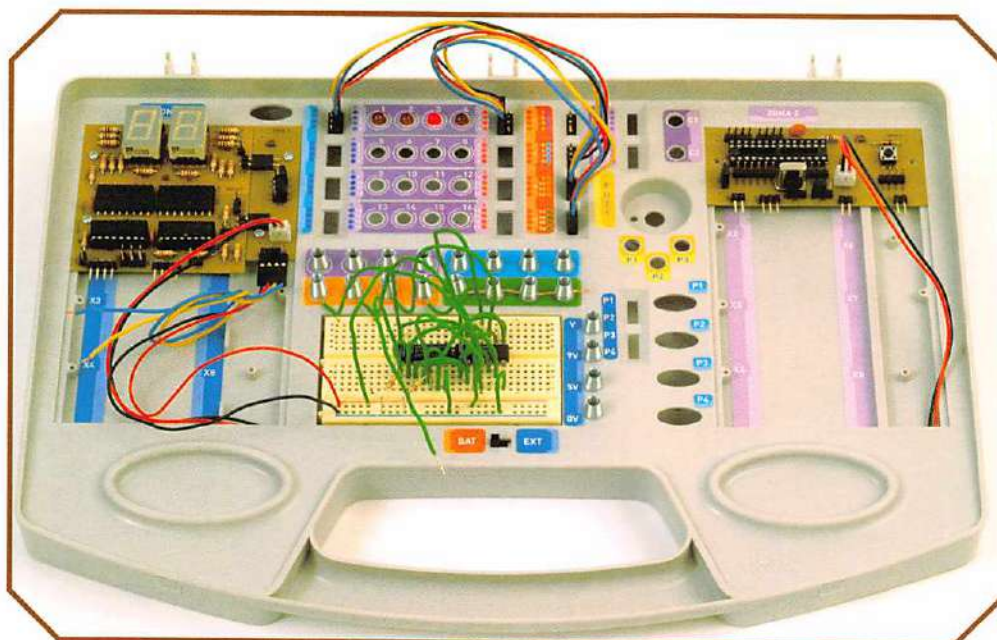
U1	Circuito integrato 4001
U2	Circuito integrato 4093
R5, R6	Resistenza 100 K (marrone, nero, giallo)



Collegando A e B contemporaneamente si illumina il LED 4.

### Prova di funzionamento

Il montaggio di questo circuito risulta laborioso ed è consigliabile verificarlo prima di collegare l'alimentazione, in quanto, anche se si utilizzano pochi componenti, ha molti fili di collegamento. Collegando l'alimentazione i contatti A e B devono essere aperti, ovvero applichiamo la combinazione 00 e si deve illuminare il LED 1, se colleghiamo solamente A si accenderà il LED 3, se colleghiamo solamente B brillerà il LED 2 e se eseguiamo entrambi i collegamenti contemporaneamente si illuminerà il LED 4.



Vista generale del laboratorio con l'esperimento. Con A collegato e B scollegato si illumina il LED 3.



## Bistabile RS sincrono

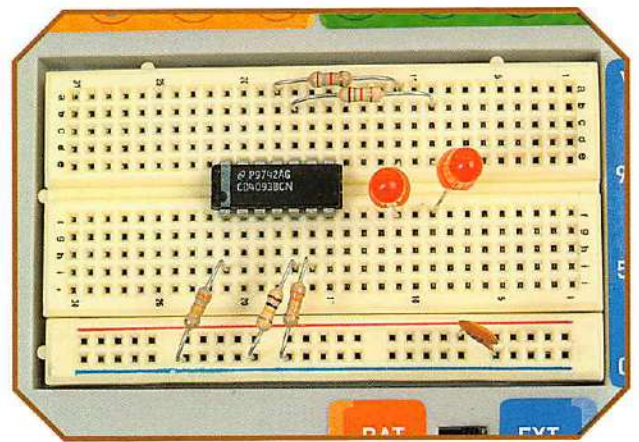
In questo esperimento si utilizza un ingresso di convalida per fare in modo che il dato presente sull'ingresso R, o su quello S, sia considerato solamente al momento opportuno. Ci sono due ingressi di segnale, SET e RESET, e un altro di convalida che indicheremo come T.

### Il circuito

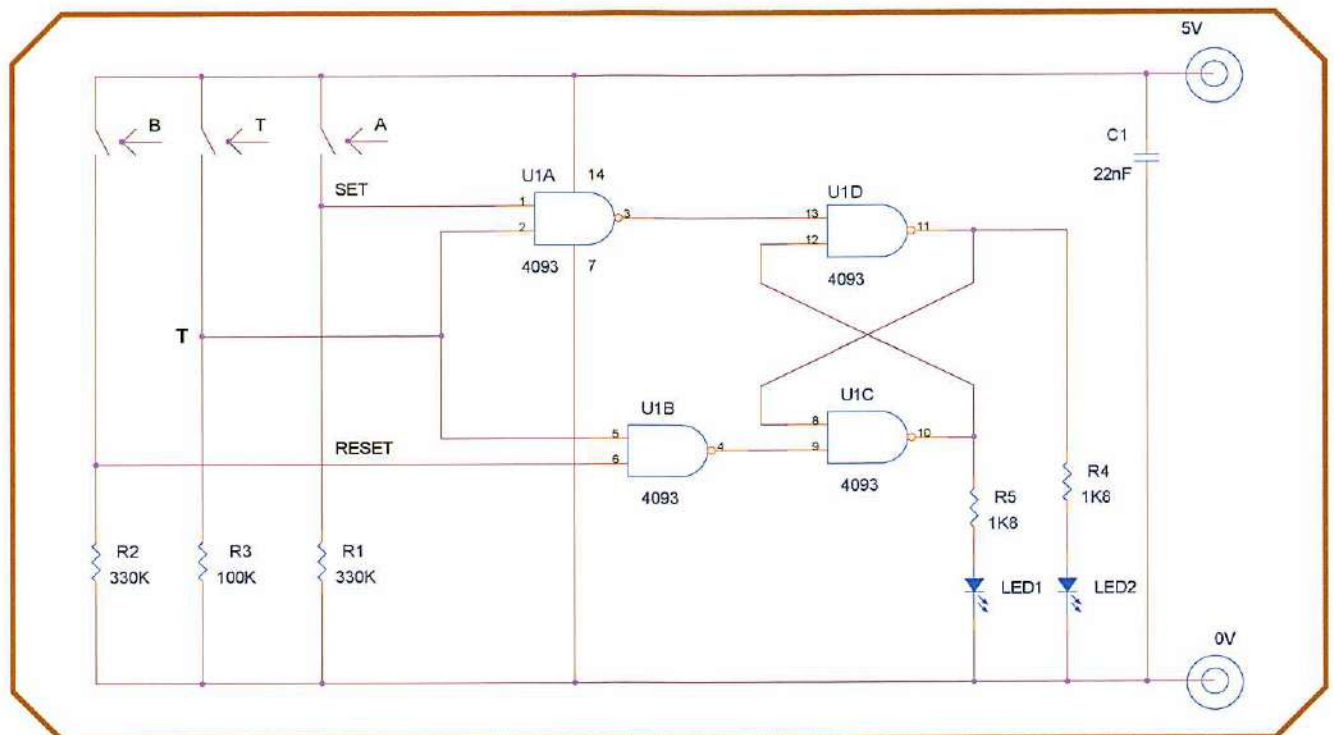
Se osserviamo lo schema vediamo che è composto da un bistabile RS realizzato con due porte NAND, U1D e U1C, con la particolarità che gli ingressi del circuito non sono applicati direttamente a queste porte che formano il bistabile. Tra ogni ingresso del circuito e il bistabile è stata interposta una nuova porta, U1A per l'ingresso di SET, e U1B per quello di RESET.

Il collegamento A si utilizza per il SET, quello B per il RESET e quello T per la convalida.

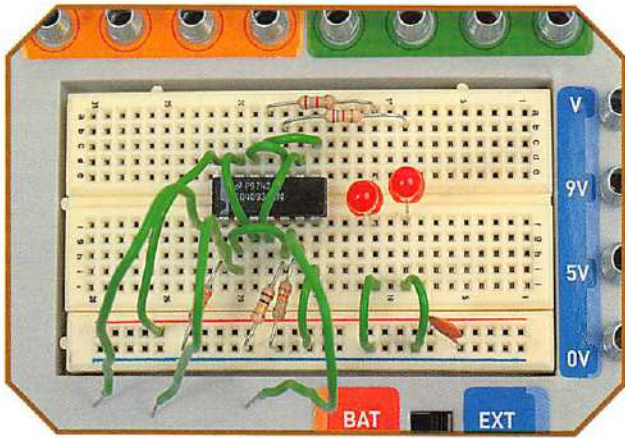
Lo schema appare più semplice dei prece-



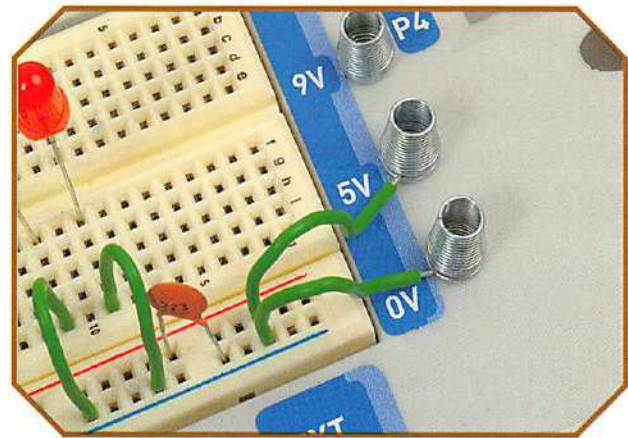
Componenti installati sulla scheda Bread Board.



Bistabile RS sincrono.



Dopo aver montato i componenti si esegue il cablaggio.



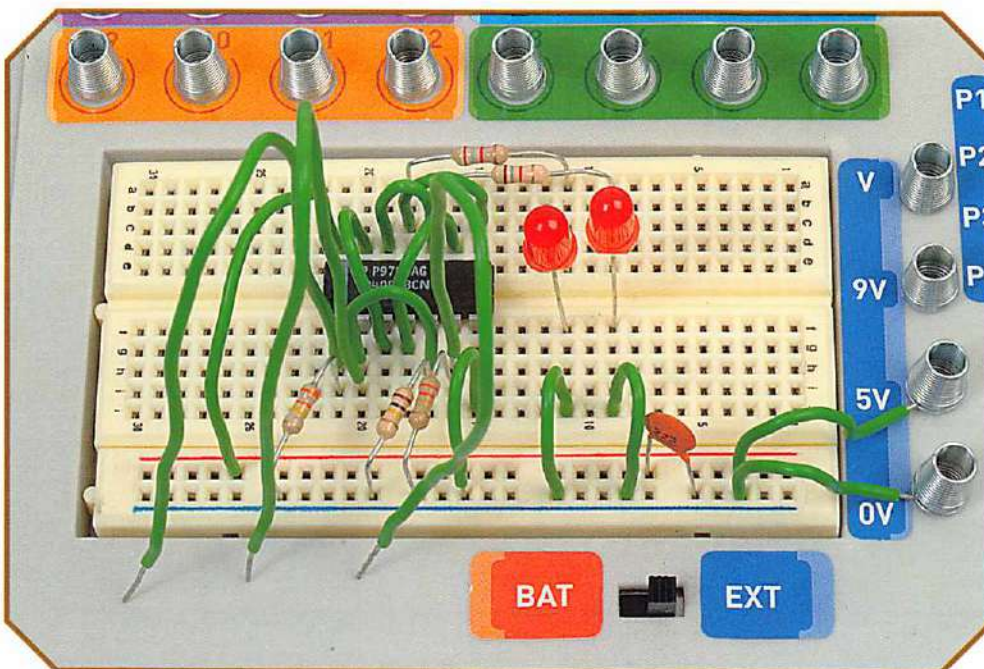
Collegamento dell'alimentazione alle molle 0 V e 5 V.

denti, dato che per l'alimentazione sono state rappresentate solamente le due molle 0 V e 5 V. Come possiamo vedere, per la rappresentazione di queste molle si utilizza il simbolo delle due circonferenze concentriche.

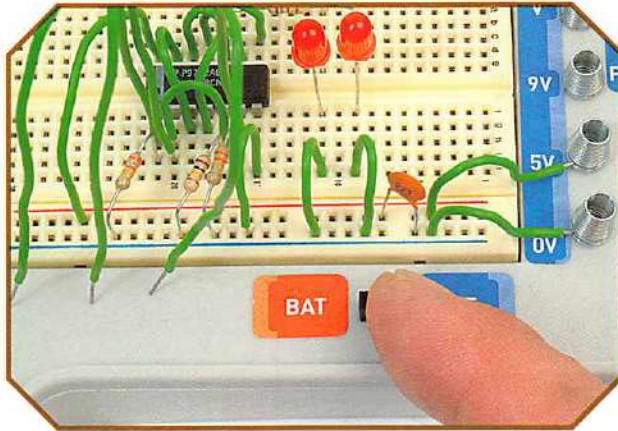
Nel circuito viene inserito un elemento nuovo, il condensatore C1, che è collegato direttamente all'alimentazione; si tratta di un tipico condensatore di disaccoppiamento dell'alimentazione. Come regola generale, tutti i circuiti integrati devono avere sull'ingresso, e molto vicino al loro terminale di alimentazio-

ne, un condensatore di disaccoppiamento, non solo per filtrare i disturbi dell'alimentazione che arrivano dagli altri circuiti, ma anche per evitare che i disturbi generati dall'integrato stesso si possano propagare agli altri circuiti. In molti casi, nei circuiti di prova alimentati a batterie e con un solo circuito integrato, questo condensatore viene ommesso, però la maggior parte delle volte è indispensabile per il corretto funzionamento dei circuiti stessi.

Il LED2 testimonia lo stato dell'uscita del bistabile, terminale 11 dell'integrato, mentre il



Fili per simulare i collegamenti A, B e T.



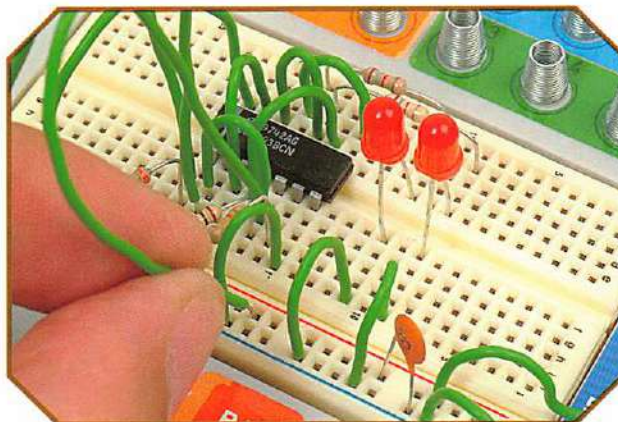
*Commutatore di alimentazione, posizione BAT per alimentare con le batterie.*

LED1 rappresenta l'uscita invertita; l'accensione dei due LED non può essere contemporanea, soltanto uno dei due può rimanere illuminato.

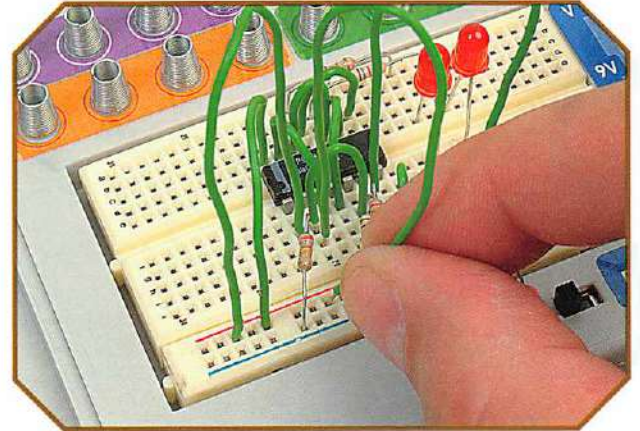
## Montaggio

A questo punto il montaggio dell'esperimento deve risultare relativamente facile, così come i collegamenti dell'alimentazione; è sufficiente unire i corrispondenti terminali della scheda Bread Board con i terminali delle mole 0 V e 5 V, posizionati nelle vicinanze, per ottenere i 4,5 volt dell'alimentazione a pile (saranno 5 V quando utilizzeremo l'alimentatore).

I fili che si utilizzano per eseguire i collegamenti A e B rappresentano rispettivamente i collegamenti di SET e RESET del circuito. Il filo



*Con T scollegato, anche se si collegano A o B, il circuito non deve reagire.*



*Bisogna collegare momentaneamente T per convalidare il collegamento realizzato precedentemente su A.*

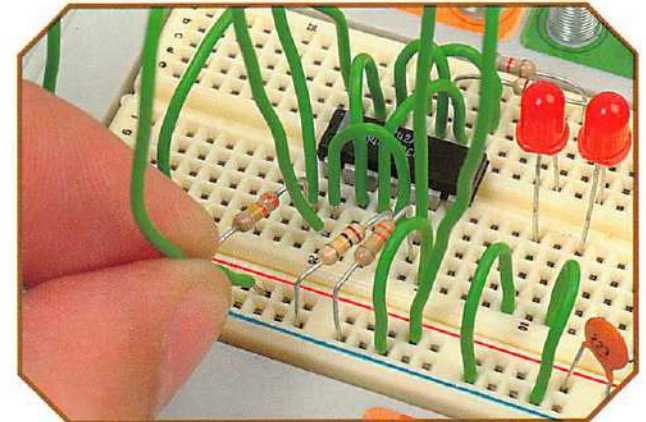
collegato a T rappresenta l'ingresso di convalida.

Tutti questi ingressi rimangono a livello basso quando non sono collegati, forzati dalle resistenze di "pull-down" R1, R2 e R3. È sufficiente quindi inserire un circuito integrato 4093 sulla scheda Bread Board, le cinque resistenze indicate e i due LED, facendo attenzione alla polarità degli stessi.

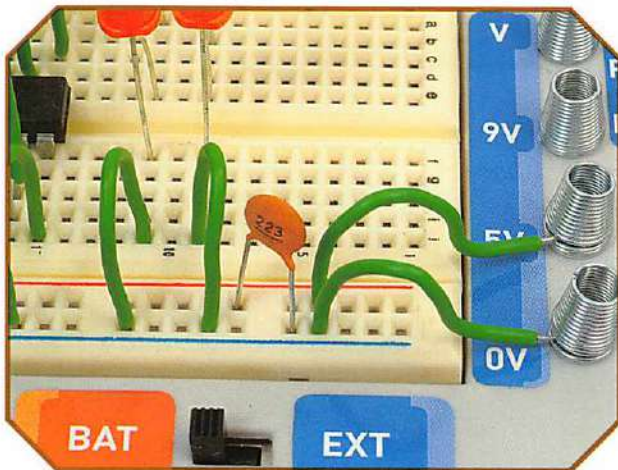
## Prova di funzionamento

Prima di collegare l'alimentazione bisogna verificare tutto il lavoro svolto, specialmente quei collegamenti dei fili che sono facili da confondere.

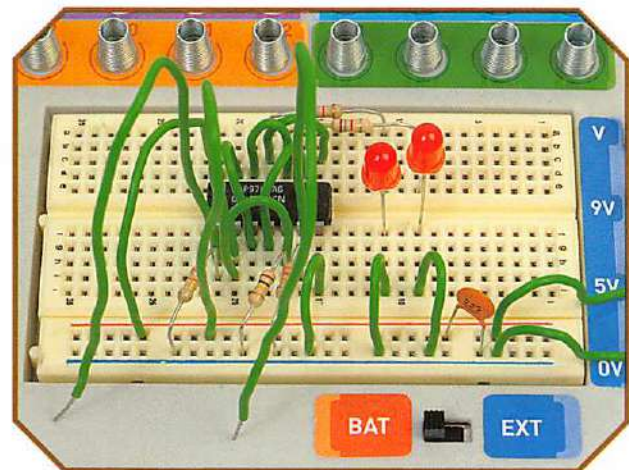
Dopo aver verificato se il montaggio è corretto bisogna portare il commutatore sulla posizione BAT per fare in modo che la corren-



*Collegando momentaneamente T con B collegato, il LED cambia.*



*C1 è un condensatore di disaccoppiamento dell'alimentazione.*



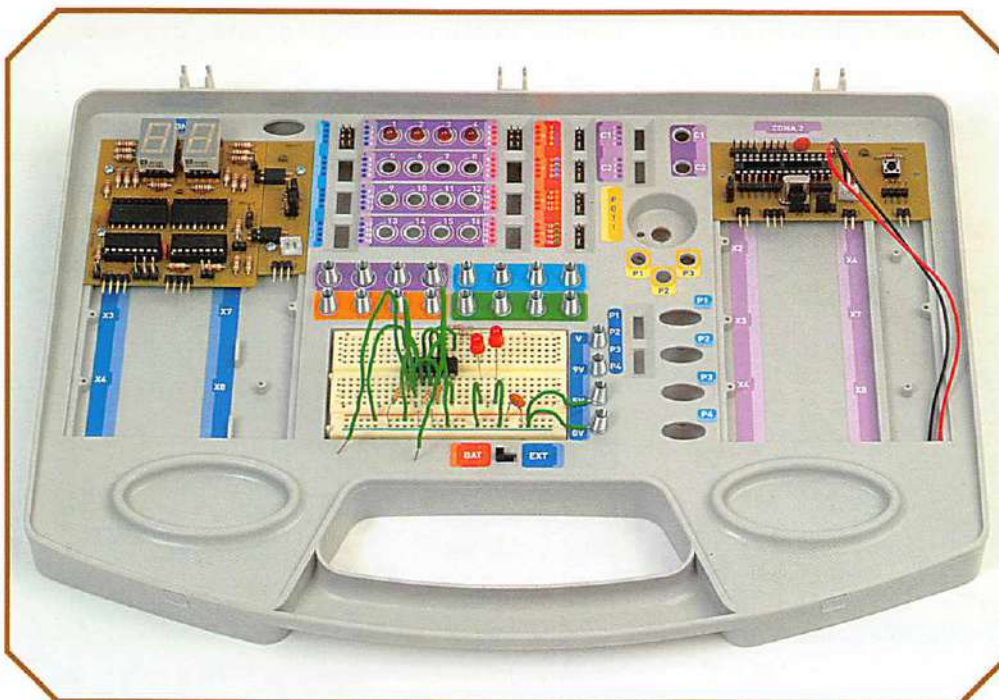
*Con T collegato in modo permanente il circuito funziona come un bistabile RS asincrono.*

#### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
R1, R2	Resistenza 330 K (arancio, arancio, giallo)
R3	Resistenza 100 K (marrone, nero, giallo)
R4, R5	Resistenza 1K8 (marrone, grigio, rosso)
LED 1, LED 2	Diodo LED rosso

te di alimentazione passi dalle pile del porta-batterie fino ai terminali 0 V e 5 V.

Uno dei due LED si deve illuminare e, se colleghiamo per un attimo A o B, mai entrambi contemporaneamente, lo stato dell'uscita non deve cambiare. Il cambio è possibile solamente se, dopo aver eseguito il collegamento, si attiva T per un attimo come convalida. Per accendere l'altro LED e spegnere quello acceso in questo momento, si cambia il collegamento al positivo passando da A a B, oppure da B ad A; l'uscita cambia solamente quando si collega T per un attimo.



*Vista generale del laboratorio con l'esperimento.*





## Astabile con due porte

In questa prova ci eserciteremo con gli astabili costruiti con le porte AND a due ingressi del circuito integrato 4093, utilizzate come porte invertenti.

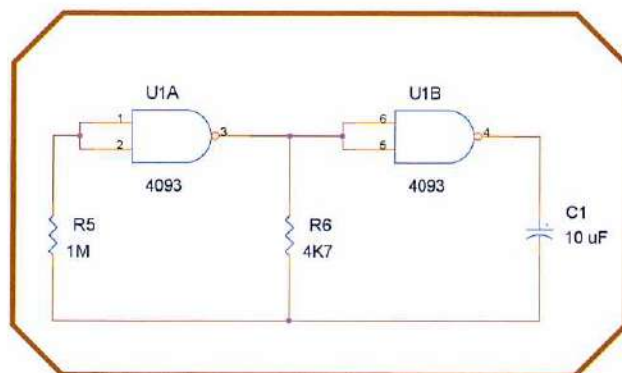
### Il circuito

L'astabile è formato dalle porte U1A e U1B, le resistenze R5 e R6 e il condensatore C1.

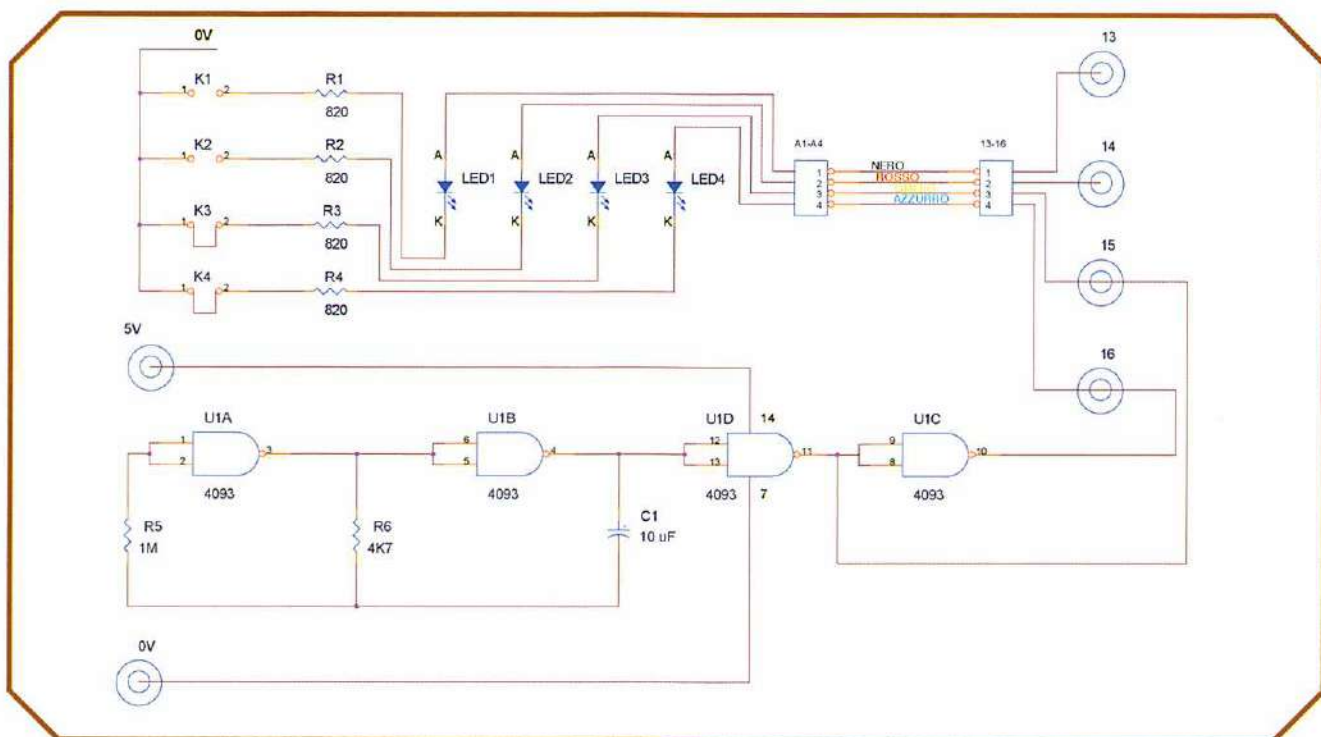
Il circuito sembra apparentemente molto complicato, tuttavia sono solamente stati aggiunti gli elementi che permettono di verificare il funzionamento del circuito stesso. L'uscita del circuito si prende dalla porta U1B, terminale 4 del circuito integrato, si porta all'ingresso di una porta invertente, formato da U1D, e l'uscita di questa a sua volta è mandata alla porta rimanente U1C.

Le uscite di queste ultime due porte, terminali 11 e 10 di U1 rispettivamente, sono utilizzate per illuminare, quando sono attive, alcuni diodi LED, precisamente quelli installati sul

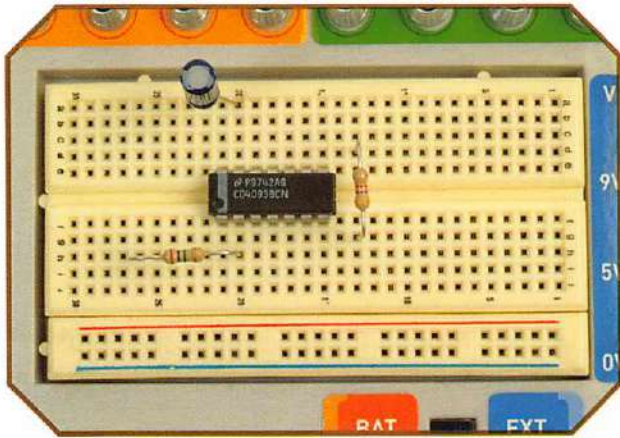
pannello frontale del laboratorio, identificati come LED3 e LED4. Le resistenze di polarizzazione di questi due LED fanno parte del circuit-



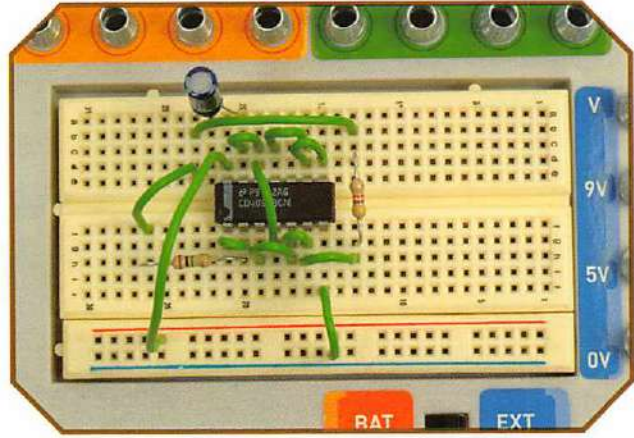
Componenti di base dell'astabile.



Astabile con due porte; segnale simmetrico.



Componenti installati sulla scheda Bread Board.



Cablaggio interno.

to stampato DG11 e non sono visibili, tantomeno sono riportate sulla lista dei materiali, perché su questa lista vengono indicati solamente i componenti montati sulla scheda Bread Board.

## Montaggio

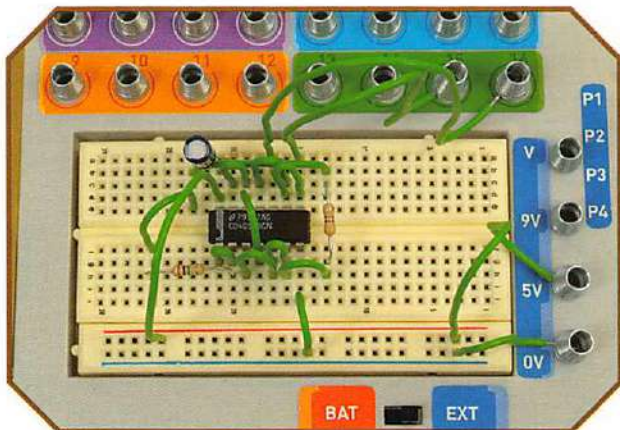
Il montaggio è piuttosto semplice e non comporta alcuna difficoltà particolare. Si realizza nel modo abituale, seguendo lo schema e aiutandoci con le figure. Bisogna fare attenzione al montaggio del circuito integrato, alla polarità del condensatore elettrolitico e, in generale, a tutti i collegamenti.

Per fornire l'alimentazione negativa al LED3 e al LED4 utilizzati nell'esperimento, è necessario collegare con due ponticelli i terminali K3 e K4 al terminale dell'alimentazione, che è a lato di ognuno di essi.

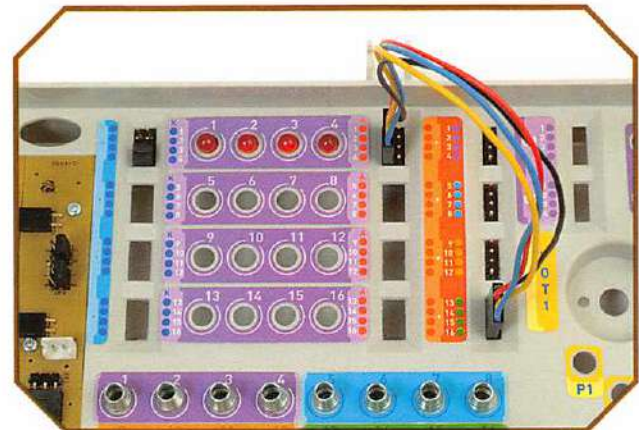
Dovremo anche utilizzare un cavetto a quattro fili, terminato su dei connettori a quattro vie, per stabilire il collegamento tra i terminali degli anodi dei LED A1, A2, A3 e A4 e i terminali dal 13 al 16 di accesso alle molle; il collegamento alla scheda Bread Board si realizza tra i terminali 11 e 10 del circuito integrato 4093 e le molle 15 e 16 rispettivamente.

## Alimentazione

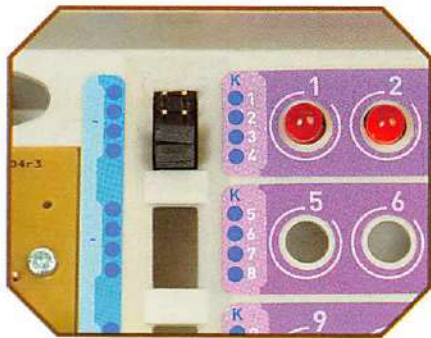
I collegamenti dell'alimentazione si realizzano direttamente sui terminali 5 V e 0 V, il negativo dei LED è permanentemente collegato alla fila di 16 terminali, raggruppati di quattro in quattro, identificati con il colore azzurro e il segno meno, posizionati ognuno a lato di ciascun collegamento del catodo, per facilitare il collegamento tra gli stessi con i ponticelli inseribili.



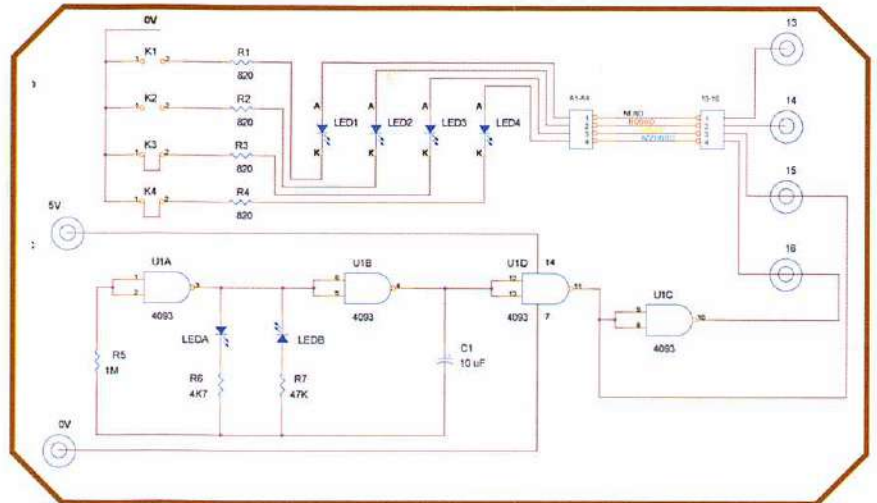
Collegamenti alle molle.



Collegamenti con il cavetto tra A1-A4 e 13-16.



Dettaglio dei due ponticelli che si utilizzano per alimentare i negativi dei LED 3 e 4.



## Prova

*Astabile con porte; segnale asimmetrico.*

Prima di collegare l'alimentazione dobbiamo rivedere tutto il lavoro svolto e, dopo esserci assicurati che è corretto, verificheremo lo stato delle tre pile del portabatterie situato sotto la zona 1, prima di spostare il commutatore sulla posizione BAT.

Il circuito deve iniziare a oscillare e i LED 3 e 4 si illumineranno in modo alternativo, entrambi con lo stesso tempo. Se si desidera che il lampeggio dei LED sia più lento si può provare con valori superiori, ad esempio 47 K e 100 K.

## Circuito asimmetrico

Con una piccola modifica possiamo ottenere che il tempo di accensione dei due LED sia diverso. Se osserviamo lo schema del circuito modificato affinché sia asimmetrico, potremo

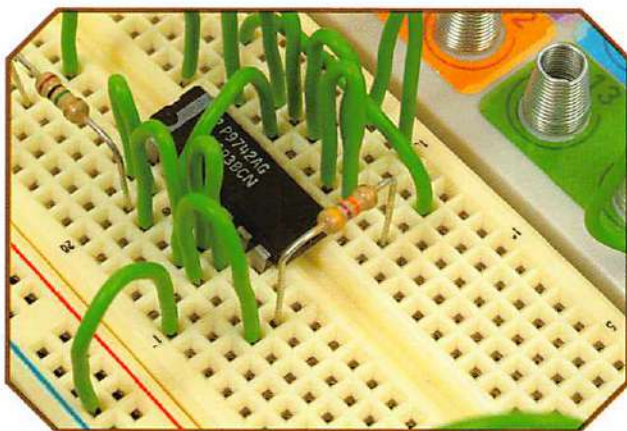
### LISTA DEI COMPONENTI

Circuito segnale simmetrico

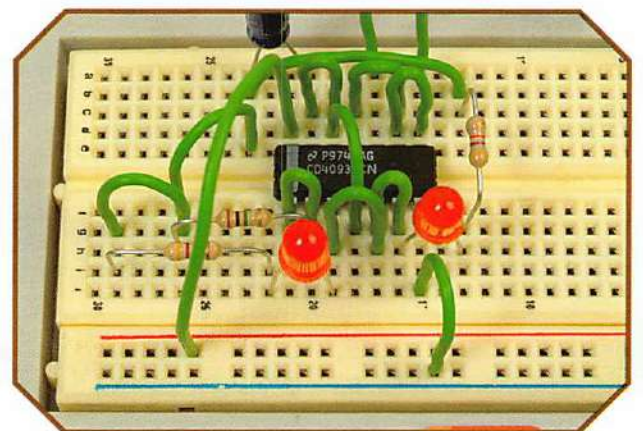
- U1 Circuito integrato 4093
- R5 Resistenza 1M (marrone, nero, verde)
- R6 Resistenza 4K7 (giallo, viola, rosso)
- C1 Condensatore 10µF elettrolitico

Circuito segnale asimmetrico

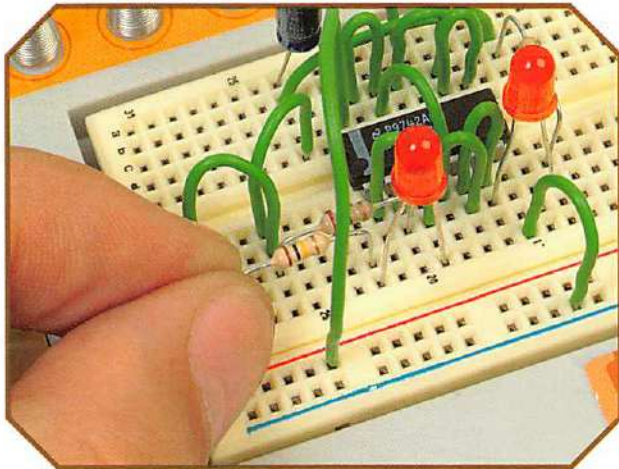
- U1 Circuito integrato 4093
- R5 Resistenza 1M (marrone, nero, verde)
- R6 Resistenza 4K7 (giallo, viola, rosso)
- R7 Resistenza 47K (giallo, viola, arancio)
- C1 Condensatore 10µF elettrolitico
- LED 1, LED 2, Diodo LED rosso



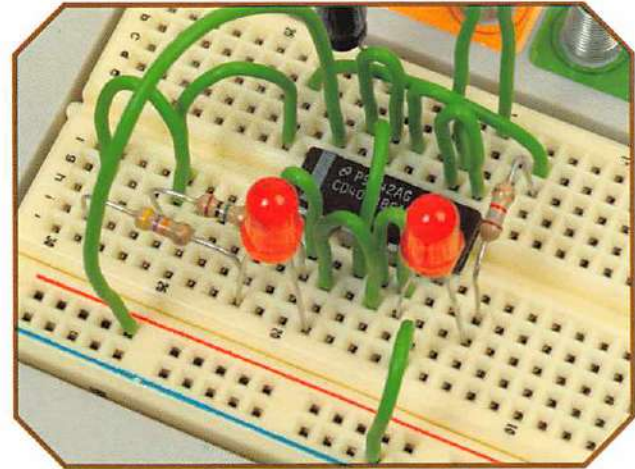
Se aumentiamo il valore di R6 si abbassa la frequenza.



Circuito modificato per ottenere un segnale asimmetrico.



In questa prova aumentiamo il valore di R7 a 100 K.



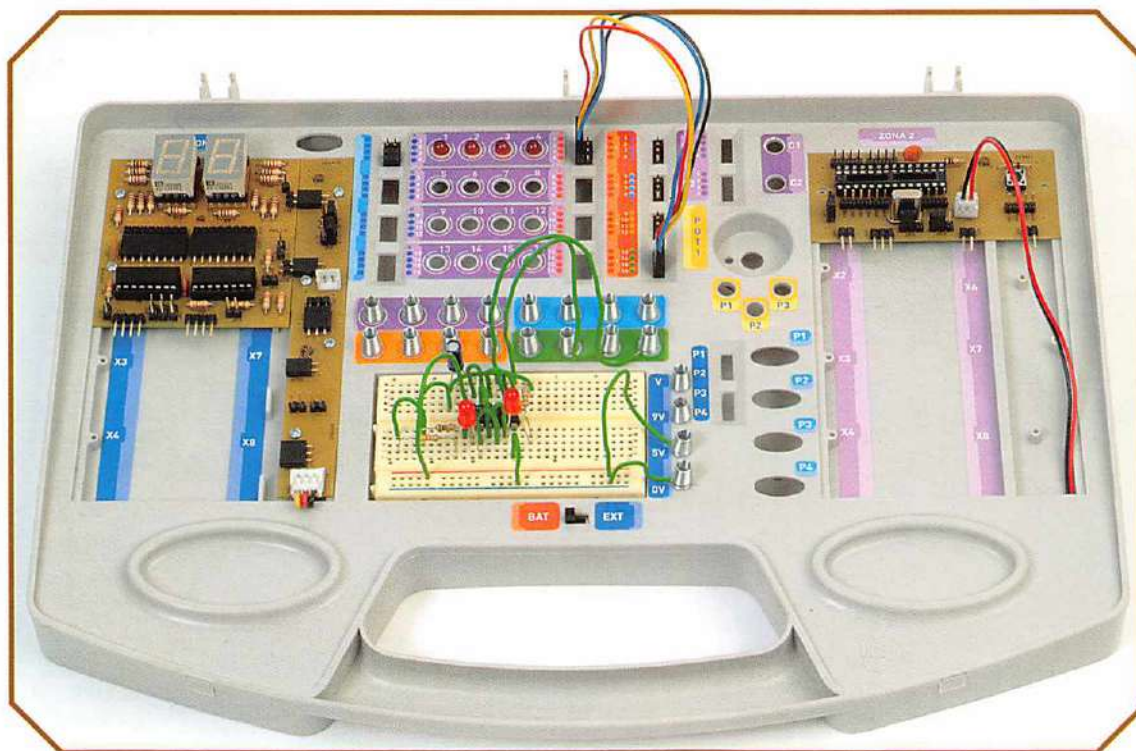
Qui diminuimo R6 a 1K8.

vedere che sono stati aggiunti i LED e una resistenza, la R7, di valore differente dalla R6. Questa differenza di valore tra R6 e R7 determina la differenza dei tempi di illuminazione tra un LED e l'altro.

### Prova del circuito modificato

I LED A e B inseriti sulla scheda Bread Board si utilizzano per fare in modo che la corrente

di carica e quella di scarica del condensatore C1 seguano percorsi diversi. Non è importante che si illuminino però è molto utile, perché potremo farci un'idea della corrente che circola attraverso i LED stessi. Possiamo provare con valori diversi per R6, ad esempio 1K8, e per R7, ad esempio 100 K, inoltre possiamo scambiare tra di loro le resistenze per verificare che si cambi il comportamento di un LED (LED3 e LED4) con l'altro.



Vista del laboratorio con il circuito asimmetrico.



# Indicatore temporizzato

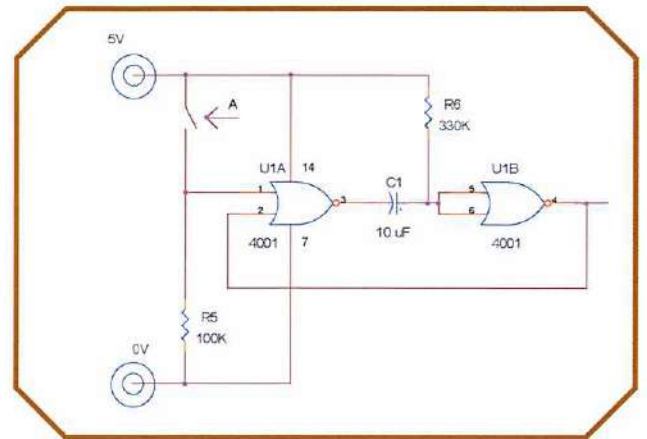
Una delle molte applicazioni dei circuiti astabili e monostabili è l'indicazione dello stato di alcuni circuiti, ad esempio gli allarmi; faremo delle prove con un circuito che utilizza entrambi i tipi di multivibratori.

## Funzionamento

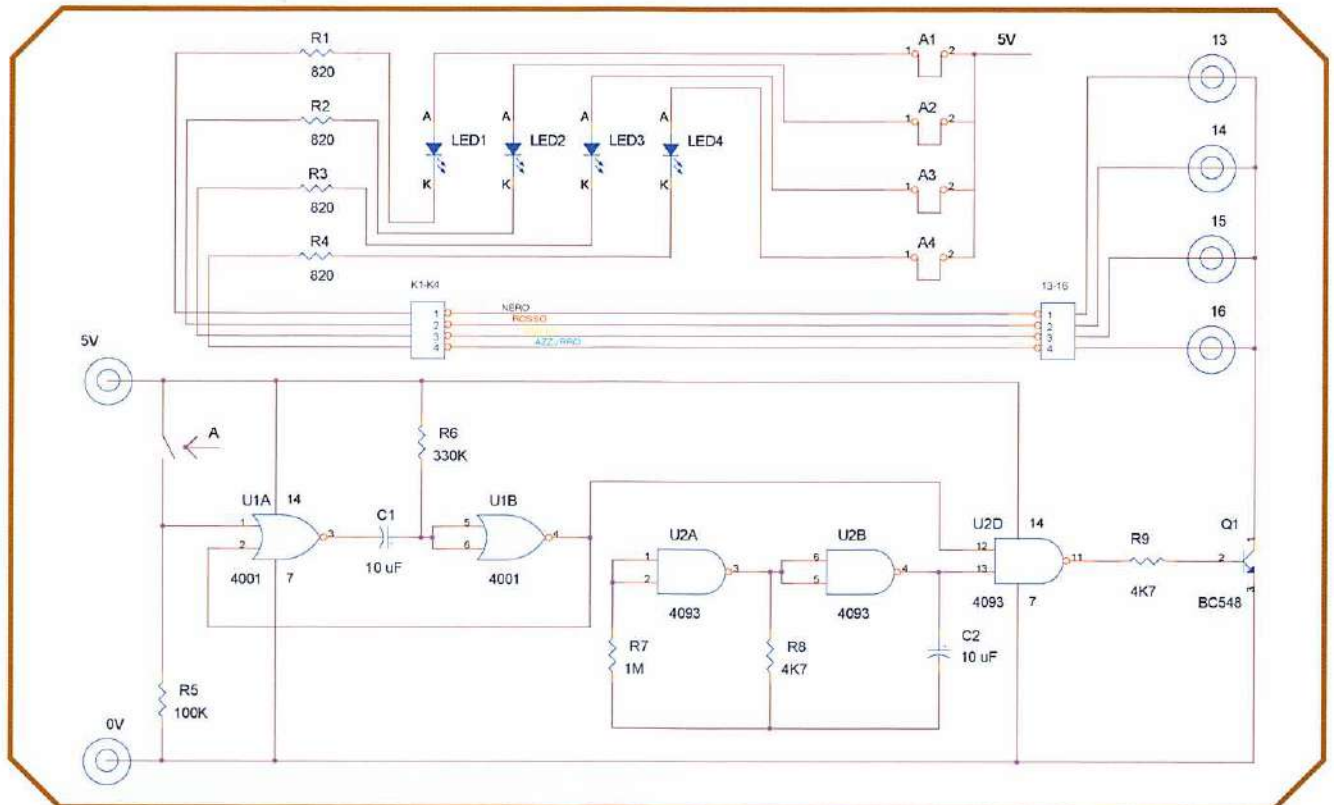
Il circuito ha un ingresso di attivazione rappresentato dal collegamento A, e quattro LED che rimangono illuminati. Quando si attiva l'ingresso A, collegando il filo per un attimo, i LED si illuminano in modo intermittente per un certo periodo, dopodiché tornano a un'illuminazione fissa fino a quando si attiverà nuovamente l'ingresso del circuito.

## Il circuito

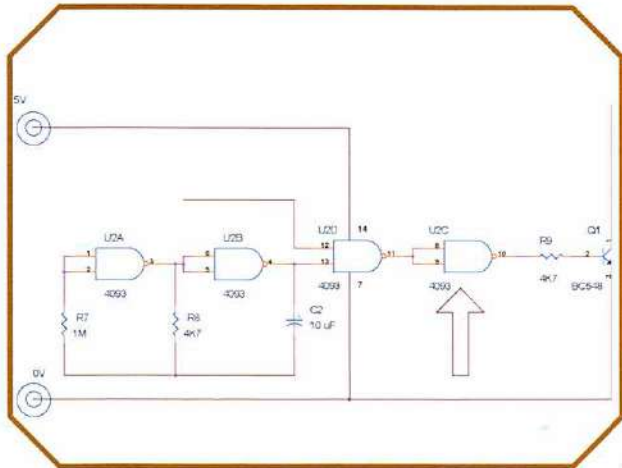
Se osserviamo lo schema vediamo che il collegamento A attiva un monostabile formato



Monostabile.



Schema del circuito dell'indicatore temporizzato.



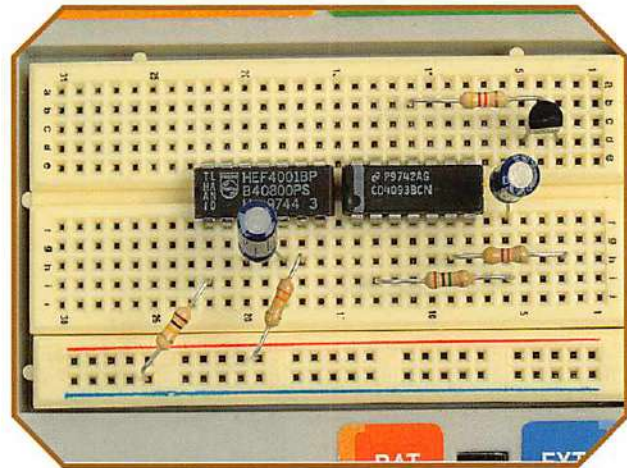
Aggiungendo una porta si inverte l'illuminazione

dalle porte U1A, U1B, il condensatore C1 e la resistenza R6. I valori di questi ultimi componenti determinano la durata di questa temporizzazione.

Nel circuito possiamo vedere anche un oscillatore astabile, formato dalle porte U2A e U2B, la cui uscita è portata a uno degli ingressi della porta U2D, e lo stato dell'altro ingresso dipende dall'uscita del temporizzatore: quando l'uscita di questa porta è uno, si polarizza il transistor Q1 in modo da illuminare contemporaneamente i LED da 1 a 4. Gli anodi dei LED sono collegati al positivo dell'alimentazione.

## Montaggio

Il montaggio si deve eseguire con molta attenzione per evitare errori. È conveniente ini-

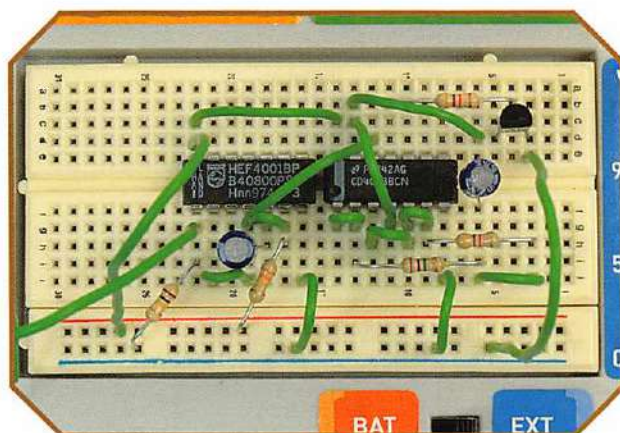


Scheda Bread Board con i componenti inseriti.

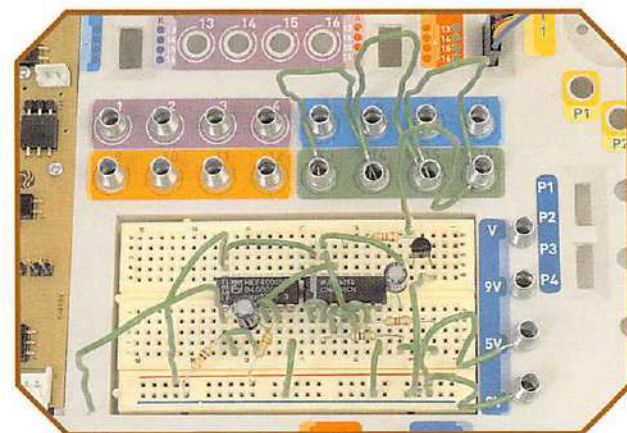
ziare inserendo i circuiti integrati e il transistor, facendo molta attenzione al suo orientamento. Dopodiché installeremo le resistenze e i condensatori elettrolitici, osservando la polarità di questi ultimi: il negativo è indicato sul contenitore degli stessi, inoltre il terminale positivo è più lungo, a meno che non sia stato tagliato per qualche motivo.

L'alimentazione del positivo dei LED si ottiene collocando i quattro ponticelli sui terminali A1, A2, A3 e A4, e i terminali situati a fianco e identificati con il simbolo "+". Questi terminali sono collegati direttamente al positivo dell'alimentazione da 5 V.

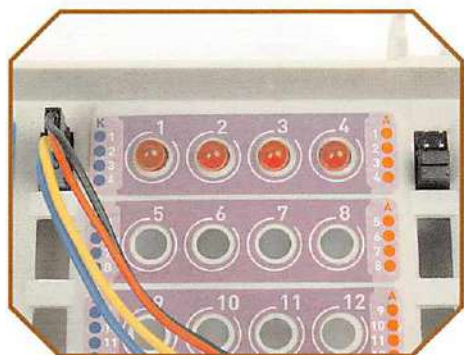
È necessario, inoltre, collegare un cavetto a quattro fili con due connettori a quattro vie, sui terminali dal 13 al 16 e sui terminali da K1 a K4; in questo modo il collegamento dei catodi di LED 1 fino a LED 4 si trasferisce alle



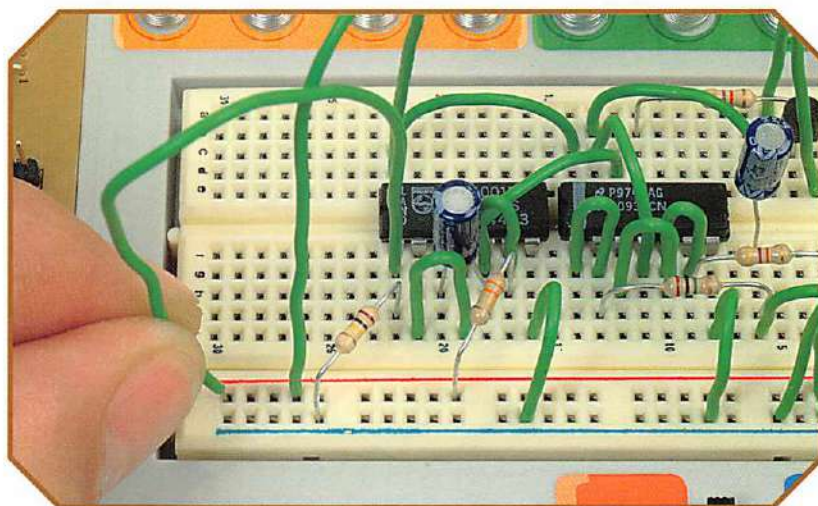
Cablaggio interno.



Collegamenti alle molle.



Dettaglio dei ponticelli che portano l'alimentazione agli anodi dei LED.



Il collegamento A si realizza con un filo.

molle 13, 14, 15 e 16. Osservando lo schema possiamo vedere che le quattro resistenze da  $820 \Omega$  limitano la corrente circolante sui LED; sono interne al circuito DG11, che contiene i quattro LED, e non sono visibili.

## Alimentazione

Il circuito si alimenta tramite le due molle di collegamento 5 V e 0 V. I collegamenti dell'alimentazione dei LED sono precablati all'interno.

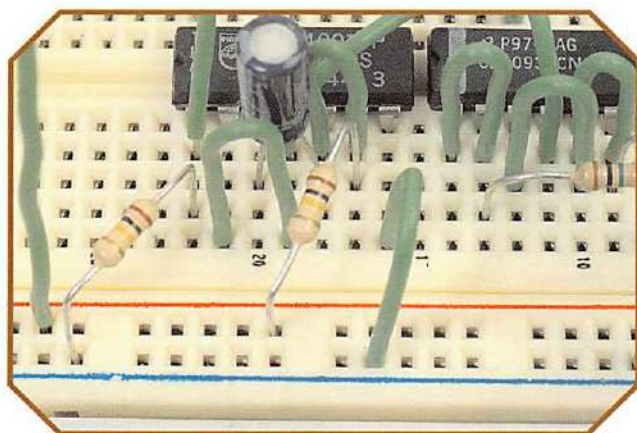
## Prova

Dopo aver verificato che il circuito è montato in modo corretto, si sposta il commutatore di alimentazione sulla posizione BAT, momento in cui i quattro LED si devono illuminare. Col-

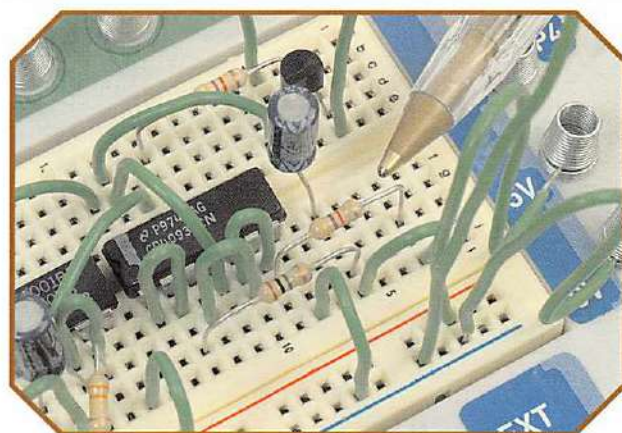
legando per un attimo A i LED si spengono e si accendono in modo intermittente per un certo periodo di tempo, dopodiché rimarranno illuminati in modo fisso.

### LISTA DEI COMPONENTI

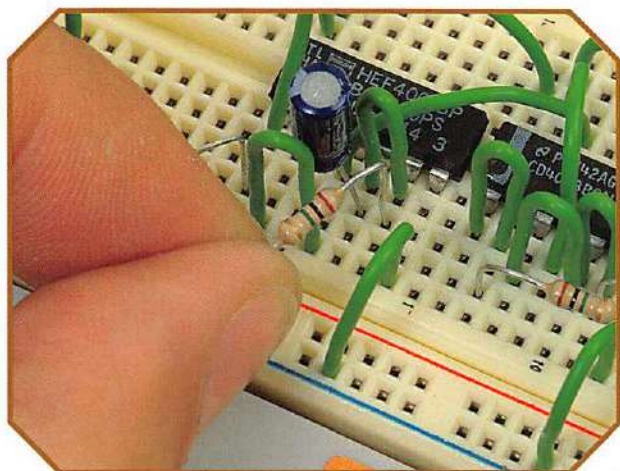
U1	Circuito integrato 4001
U2	Circuito integrato 4093
Q1	Transistor BC547 o BC548
R5	Resistenza 100 K (marrone, nero, giallo)
R6	Resistenza 330 K (arancio, arancio, giallo)
R7	Resistenza 1 M (marrone, nero, verde)
R8, R9	Resistenza 4K7 (giallo, viola, rosso)
C1, C2	Condensatore 10 $\mu\text{F}$ , elettrolitico



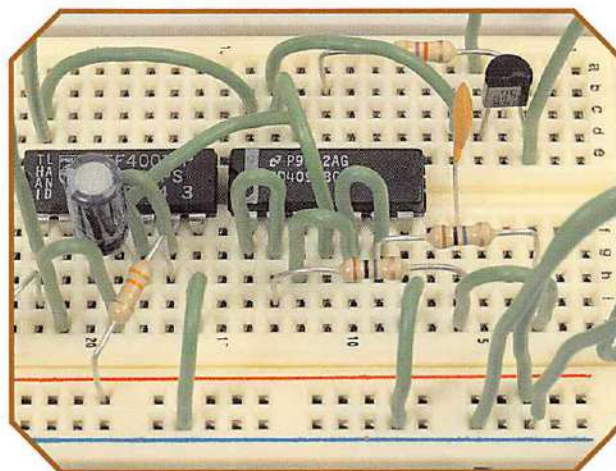
R6 determina la temporizzazione.



R8 controlla l'intermittenza dei LED.



Con R6 da 1 M  
la temporizzazione aumenta.



Con R8 da 1 M e C2 da 22 nF,  
continua a funzionare ma con un'altra frequenza.

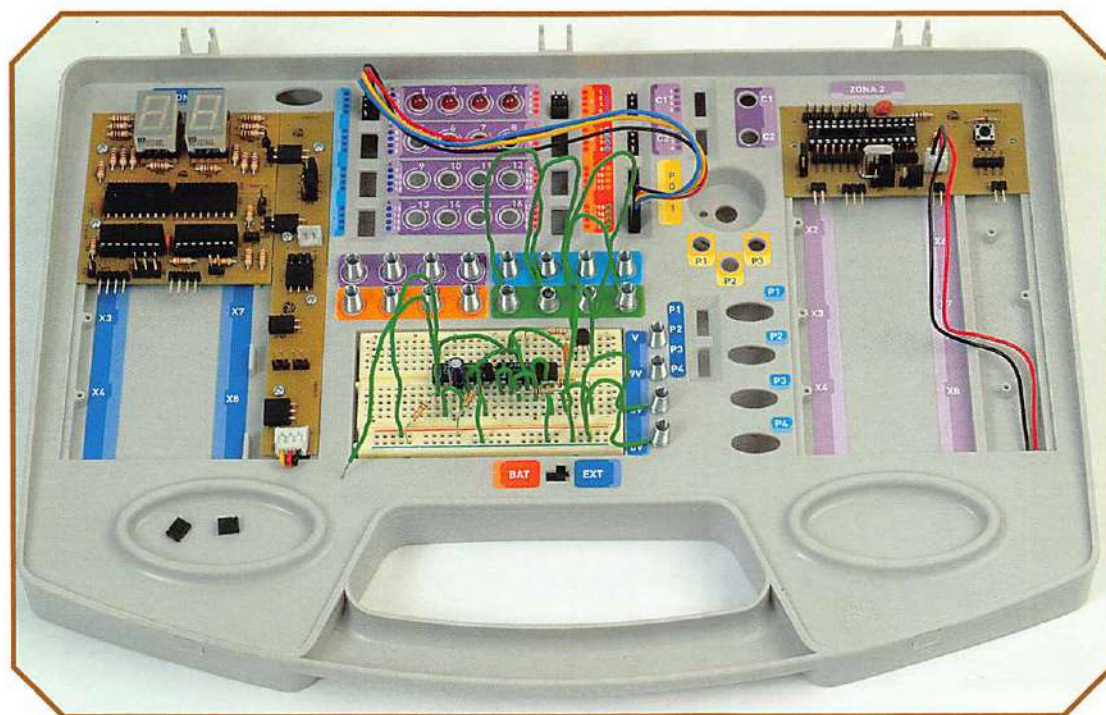
## Varianti

In questo circuito è possibile cambiare il valore di alcuni componenti, ad esempio, se diminuiamo R6 la temporizzazione diminuisce, possiamo provare con 100 K e con 1 M.

La resistenza R8 determina, insieme a C2, la cadenza o frequenza dell'intermittenza, possiamo ad esempio, tornare a porre R6 da 330K, passare R8 a 1 M e utilizzare per C2 la

capacità di 22 nF. Potremo osservare delle variazioni nella cadenza dell'intermittenza.

Possiamo provare anche un'altra variazione, inserendo una porta collegata come invertente tra l'uscita della porta U2D e la resistenza R9. Questo cambio inverte l'illuminazione dei LED che ora rimangono spenti in stato di riposo e si illuminano solamente in modo intermittente quando si attiva l'ingresso A.



Vista generale  
dell'esperimento.





# Primo esperimento con PIC 16F870

**D**isponiamo già della scheda DG06 con il PIC programmato. Dato che disponiamo di un programma scritto, che più tardi potremo modificare, scrivere o cancellare, è già possibile utilizzarlo per realizzare qualche esperimento. È arrivato il momento di iniziare l'esecuzione di un progetto in tutte le sue fasi: analisi, progetto, programmazione, simulazione e montaggio.

## Il microcontroller PIC 16F870

Il microcontroller PIC 16F870 ha il suo posto riservato nel laboratorio e non sarà più necessario toglierlo da lì per nessuna operazione. Potremo scrivere, cancellare, comunicare con esso mediante un PC e collegarlo a qualsiasi elemento del laboratorio utilizzando cavi di connessione e modificando i "jumper" o "ponticelli" che si trovano sulle schede DG06 e DG07.

## Ponticelli della scheda DG06

La scheda DG06 ha tre connettori su cui dovremo inserire dei jumper: JP1, JP2 e JP3. Ognuno di questi connettori possiede tre terminali che uniremo mediante un ponticello a due vie; ogni connettore, quindi, ha due combinazioni, in funzione all'inserimento del ponticello sui terminali 1 e 2 oppure su 2 e 3.

## I jumper JP2 e JP3

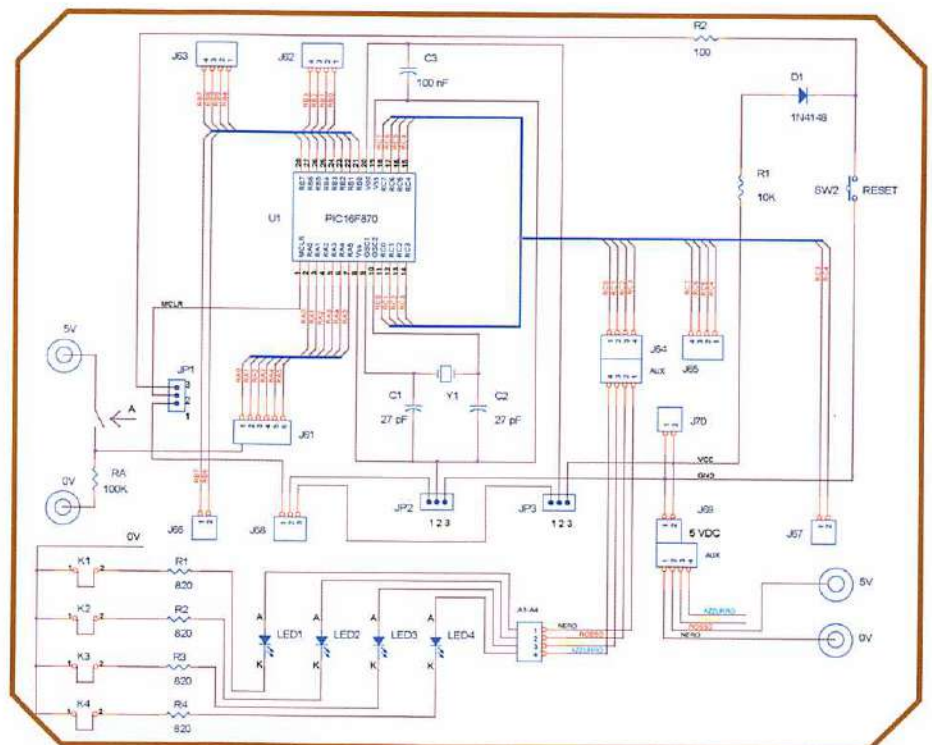
Questi ponticelli selezionano l'origine dell'alimentazione del PIC. Il PIC ha due possibili alimentazioni, una è quella normale del laboratorio che entrerà attraverso il connettore J69 (Vcc e GND), e l'altra è l'alimentazione che utilizzeremo per scrivere il PIC, che entrerà attraverso il connettore J68. Quando programiamo il PIC prendiamo l'alimentazione di-

rettamente dal PC, infatti, mediante il cavo di comunicazione, questa arriva al laboratorio.

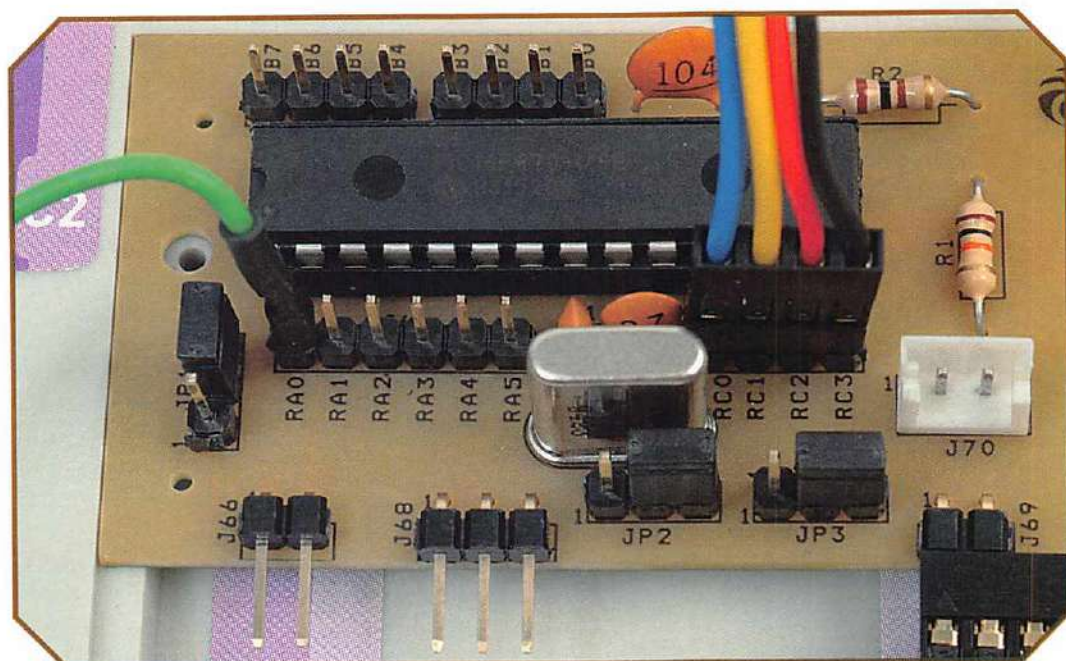
Se guardiamo lo schema elettrico e seguiamo le linee di collegamento vediamo che le linee di massa GND arrivano al JP2 e le linee del positivo o Vcc al JP3.

Il terminale 1 sullo schema elettrico è identificato sulla scheda anche con il numero 1, quindi, se vogliamo che l'alimentazione sia quella di funzionamento normale del laboratorio, come nel nostro caso, dobbiamo unire mediante un ponticello i pin 2 e 3 del connettore JP2 e i pin 2 e 3 del connettore JP3.

Osservate che il terminale 2 di questi connettori è quello comune alle possibili combinazioni e va direttamente al microcontroller.



Schema elettrico dell'esperimento.



Scheda DG06 con i ponticelli configurati per il funzionamento normale.

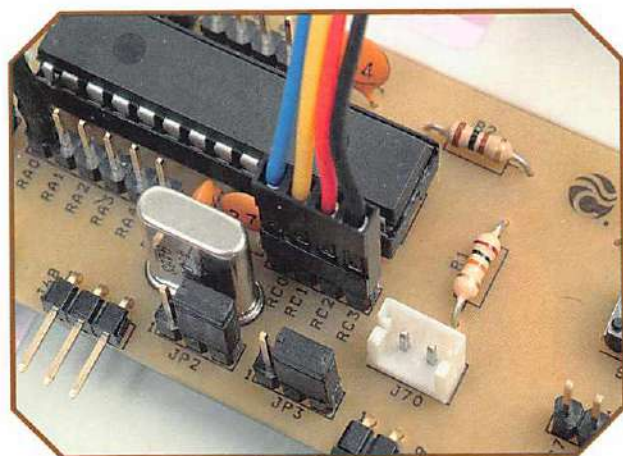
## Il ponticello JP1

Questo ponticello ha la funzione di discriminare l'ingresso al terminale MCLR dell'integrato. In questo modo, se stiamo scrivendo il PIC, prenderemo il segnale dal connettore J68 di cui abbiamo parlato in precedenza, altrimenti avremo la possibilità di eseguire il reset mediante un pulsante, come si può vedere nello schema elettrico. Quest'ultima opzione è quella normale di qualsiasi applicazione e, come per i connettori precedenti, per fare in modo che il PIC sia collegato a questo pulsante dovremo inserire il ponticel-

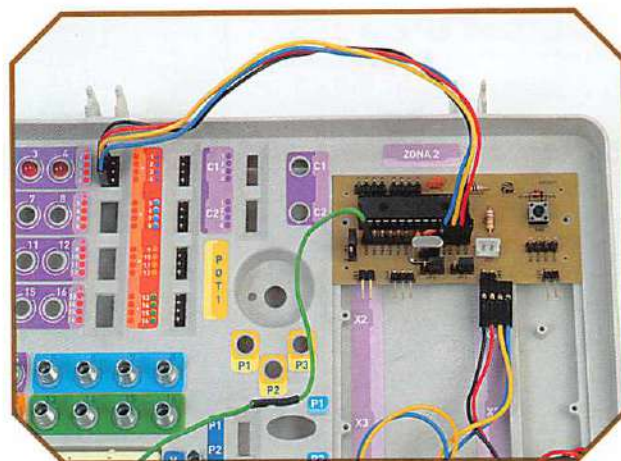
lo sui terminali 2 e 3 del JP1. È consigliabile verificare con le fotografie il posizionamento dei ponticelli. Questa è la configurazione normale, e i ponticelli cambieranno posizione solamente nel caso desiderassimo scrivere il PIC.

## Enunciato del progetto

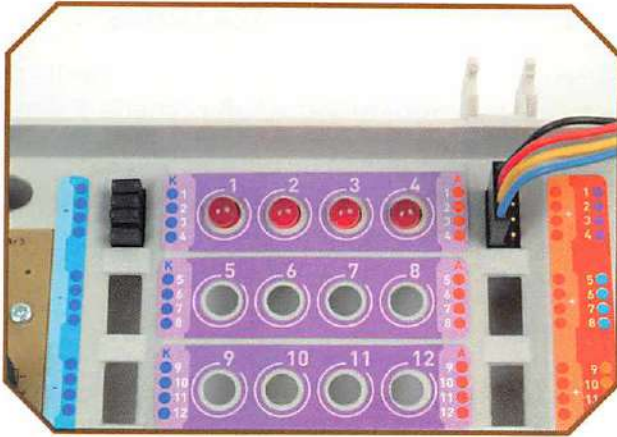
Vogliamo realizzare un progetto in cui si generino e si visualizzino una serie di numeri casuali (da 1 a 9 in decimale) che per il momento vedremo in formato binario, utilizzando i LED 1, 2, 3 e 4. Il conteggio si potrà fermare in



Collegamento a RC0-RC3 della porta C.



Gli anodi si collegano a RC0-RC3 con un cavetto a quattro fili.

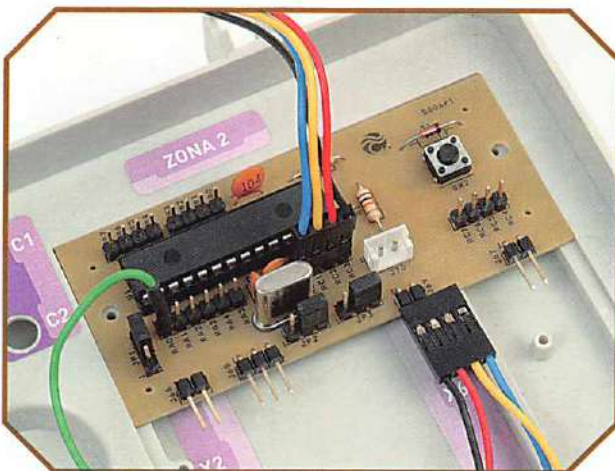


Sui catodi si collegano i quattro ponticelli.

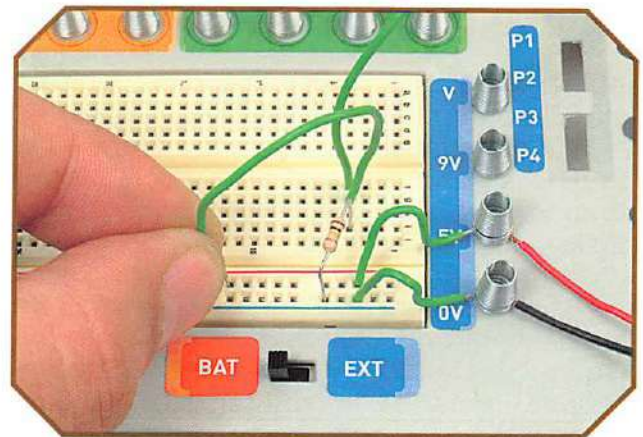
qualsiasi momento, a quel punto vedremo un numero casuale per tre secondi e, trascorso questo tempo, ricomincerà la visualizzazione sequenziale.

### Analisi del progetto

Anche se la sequenza logica, al momento di realizzare un progetto, consiste nel fare prima un'analisi e successivamente realizzare il programma e simularlo, cambiamo l'ordine, realizzando prima un possibile montaggio in modo da poter vedere un risultato finale, dato che il PIC fornito è già programmato con il programma che risolve questo progetto. In seguito spiegheremo nel dettaglio questo programma.



Il cavetto di alimentazione si collega a J69.

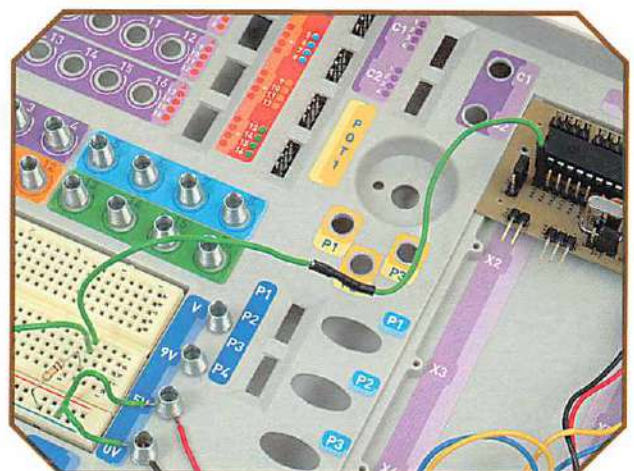


Montaggio del selettore di ingresso sulla scheda Bread Board.

### Montaggio: Visualizzazione/Uscite

Utilizzeremo i terminali da RC3 a RC0 della porta C per estrarre il segnale che controlla l'accensione dei LED, i quali rappresentano il numero in binario.

Con uno dei due cavi a 4 fili di cui disponiamo uniremo il connettore J64 (uscite RC0:RC3) della scheda DG06 agli anodi dei diodi LED, come possiamo vedere nello schema e nelle fotografie. Bisogna prestare particolare attenzione nel mantenere una corrispondenza con i colori dei fili. Se incrociamo uno dei cavetti la sequenza risulterà invertita e la visualizzazione non sarà corretta.



Collegamento di RA0 alla scheda Bread Board.



## Montaggio: Pulsante/Ingresso

Abbiamo bisogno di un ingresso per indicare al PIC quando deve mostrare il numero casuale. Questo ingresso verrà fornito tramite il pin RA0 della porta A. Quando RA0 è a "1", si vedranno i numeri in modo sequenziale. Quando RA0 torna a livello "0", verrà visualizzato il numero casuale ottenuto.

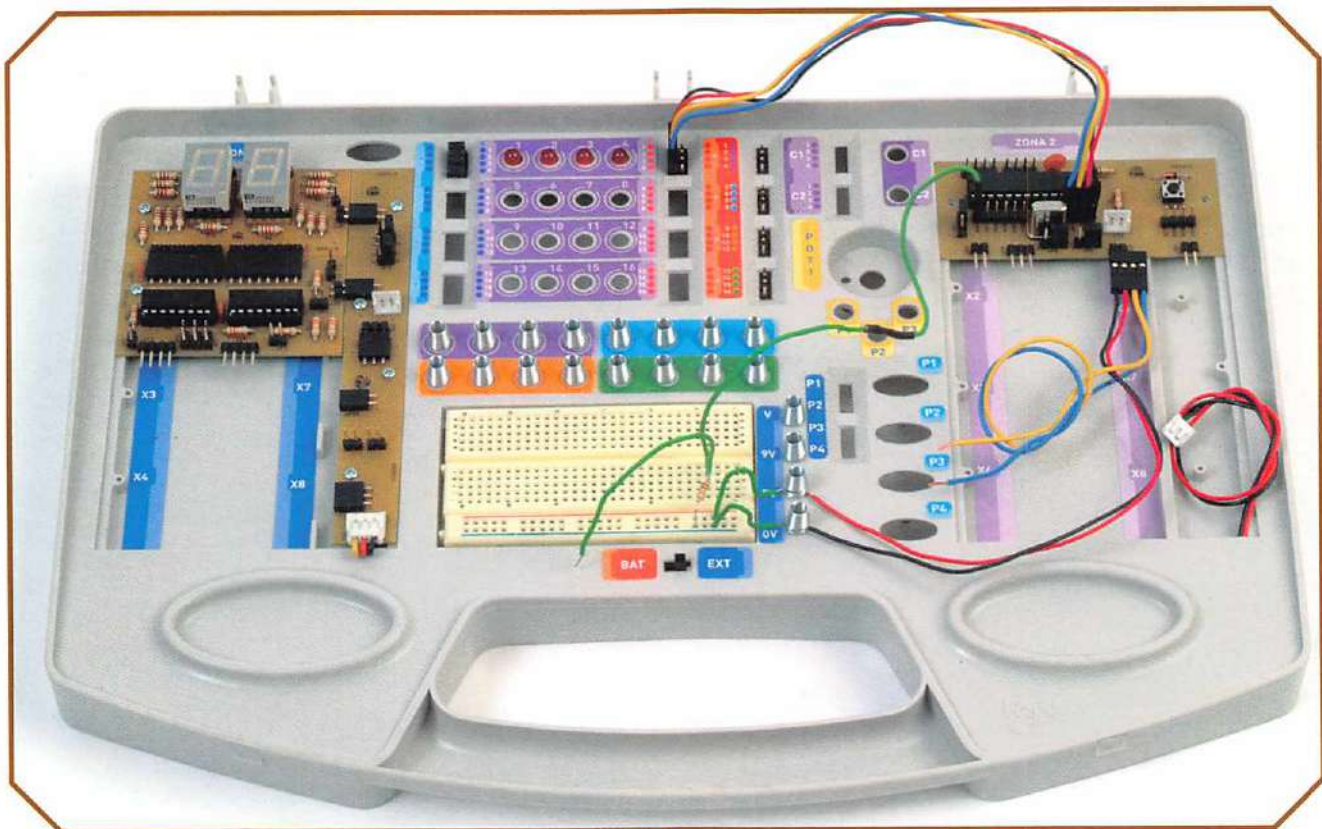
Dato che per ora non disponiamo dei pulsanti del laboratorio, utilizzeremo un filo per collegare il terminale RA0 al negativo dell'alimentazione, inserendo una resistenza da 100K, resistenza di pull-down, per assicurare uno 0 quando non è attivo il collegamento di controllo, che si esegue sul positivo dell'alimentazione. Questo collegamento di RA0 si realizza con il cavetto a un solo filo prolungato con un ulteriore pezzo di filo di collegamento. Con questo semplice montaggio dobbiamo solamente collegare il cavetto che va da un estremo della resistenza RA da 100 K alla linea del positivo della scheda Bread Board per ottenere un 1, o scollegarlo per ottenere uno 0.

## Montaggio: Alimentazione

La scheda DG06 si alimenta con le due molle 0 V e 5 V posizionate vicino alla scheda Bread Board. Per questo scopo impiegheremo un cavetto a quattro fili, di cui ne utilizzeremo solamente due. Il filo rosso verrà utilizzato per il positivo, collegandolo a 5 V, il nero per il negativo, collegandolo a 0 V, come GND.

## Risultato

Nella fotografia si può vedere il risultato finale del montaggio. Collegando l'alimentazione al connettore J69 della scheda DG06, con l'interruttore dell'alimentazione in posizione "BAT", ovvero con le batterie, otterremo una visualizzazione sequenziale di numeri con un intervallo di 0,05", e quando agiremo sul terminale RA0 avremo la rappresentazione di un numero casuale per circa 3" approssimativamente. Presto vedremo la visualizzazione diretta in decimale del numero che ora vediamo in binario.



Vista generale dell'esperimento realizzato.



# Secondo esperimento con il PIC16F870

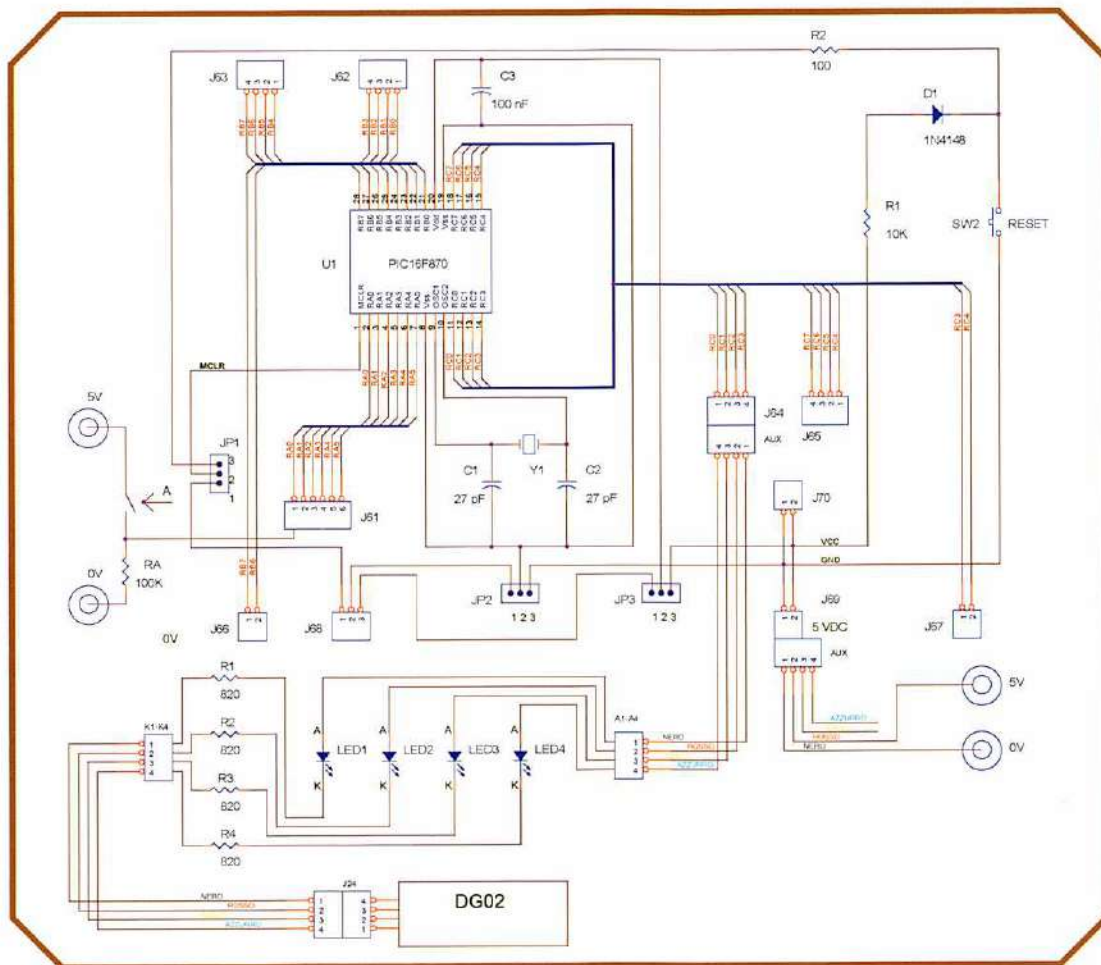
**In questa seconda prova che corrisponde al primo progetto, sappiamo già come è stato realizzato il programma. Si utilizza la stessa parte che il programma utilizzava nel primo esperimento, e le stesse uscite, terminali da RC3 a RC0 della porta C, per controllare l'accensione dei LED con cui si rappresentava il numero in binario.**

Abbiamo realizzato un montaggio che risolve il nostro progetto, ma è possibile migliorarlo?

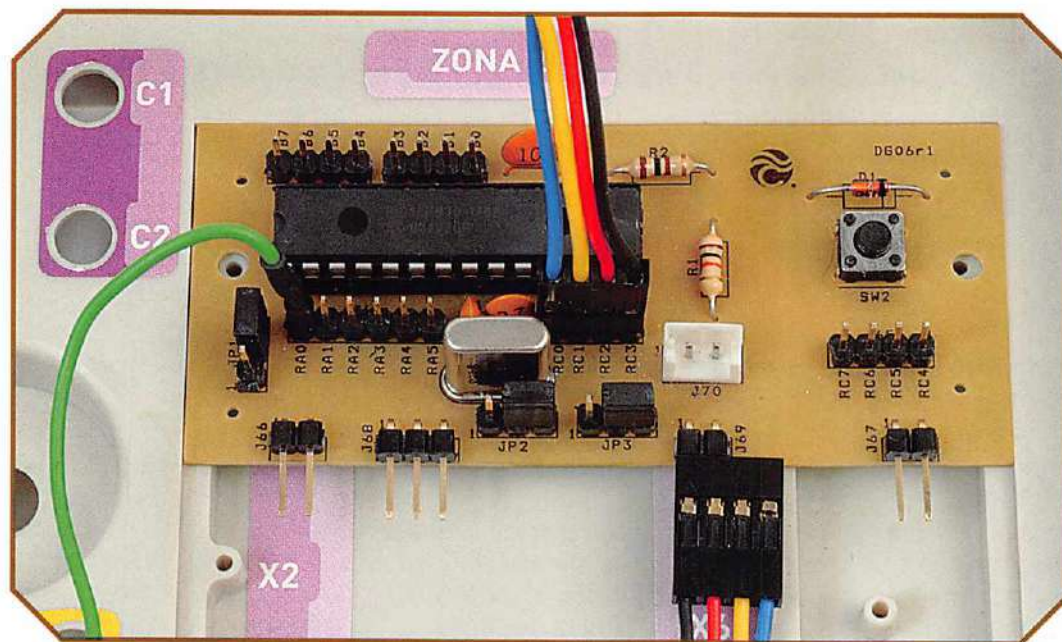
Come si può verificare questo progetto è stato risolto in modo molto semplice e relativamente economico, ma con uno scarso effetto visivo. Utilizzando un altro tipo di periferica possiamo migliorare la presentazione finale del progetto, anche se verranno penalizzati sia il costo finale che il tempo impiegato.

## Utilizzo di display a 7 segmenti

La soluzione di rappresentare i numeri dallo 0 al 9 in codice binario è corretta ma poco pratica, quindi utilizzeremo questa stessa uscita applicata all'ingresso di uno dei driver 4511 della scheda DG02, per ottenere la visualizzazione del numero direttamente sul display a 7 segmenti.



Schema elettrico corrispondente all'esperimento.



Circuito stampato DG06, con i ponticelli in posizione normale.

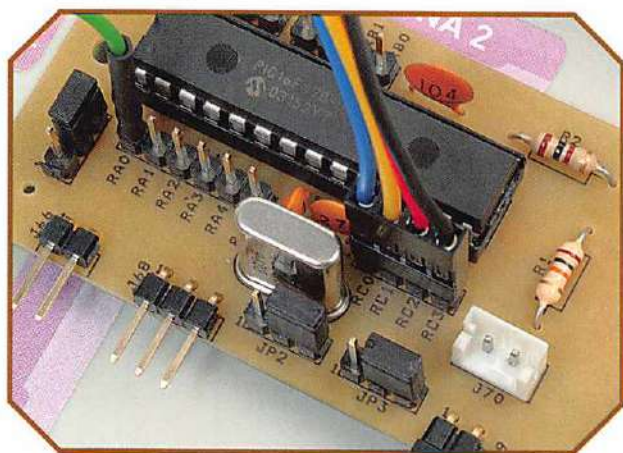
Il vantaggio di utilizzare il driver è che dal punto di vista del funzionamento del PIC è esattamente uguale al primo esperimento.

### Il lavoro con i driver

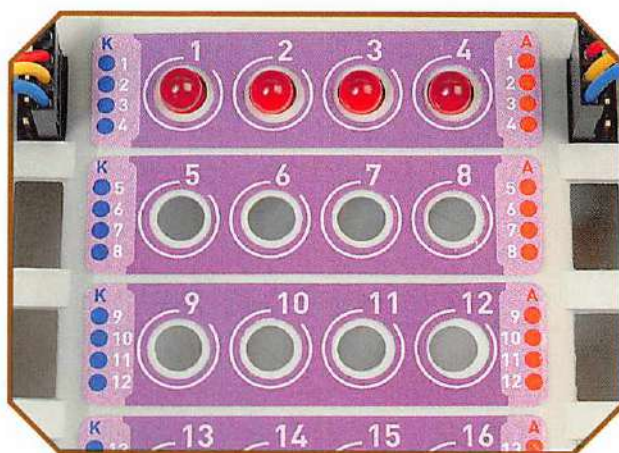
Una soluzione per vedere i numeri sul display a 7 segmenti è utilizzare i driver. L'integrato 4511 realizza una conversione interna di un numero in sistema binario al corrispondente numero in codice a 7 segmenti. In questo modo, se noi introduciamo sull'ingresso del driver il numero che vogliamo rappresentare in sistema binario, l'uscita fornirà lo stesso nu-

mero ma in formato a 7 segmenti. Per rappresentare i numeri dall'1 al 9 - con cui lavoriamo in questo progetto - abbiamo bisogno di quattro bit se lavoriamo in binario e di 7 bit se lavoriamo in codice a 7 segmenti, quindi il chip dovrà avere 4 ingressi e almeno 7 uscite.

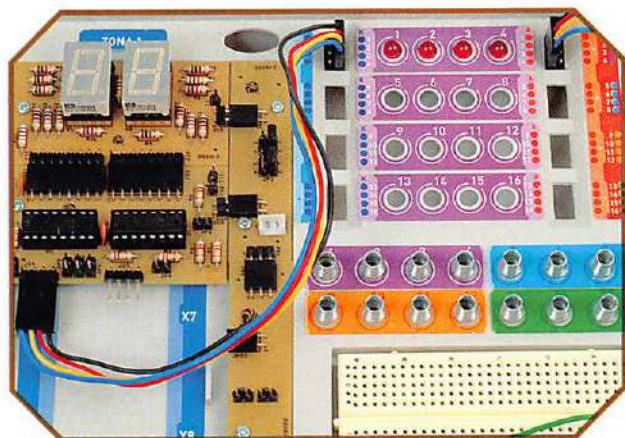
Tale soluzione è dedicata a questo esperimento, tuttavia il programma del PIC è predisposto per fare a meno del driver (ma questo corrisponde a un altro esperimento) e realizzare esso stesso le sue funzioni; in questo caso utilizzeremo la porta B come abbiamo visto nella spiegazione del progetto.



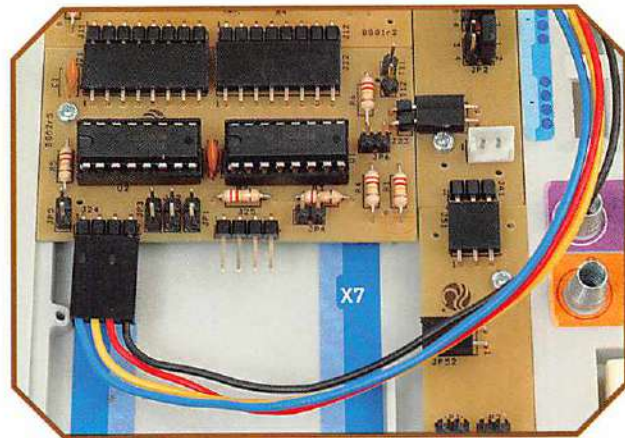
Collegamenti alla porta C di DG06.



Collegamenti del cavetto ai catodi dei LED.



Collegamenti fra i catodi dei LED e la scheda dei driver.



Dettaglio del collegamento alla scheda dei driver.

## Il montaggio

Se rispettiamo il montaggio che avevamo effettuato per il primo esperimento del circuito e colleghiamo mediante un cavetto l'altro estremo della matrice dei LED, cioè quella corrispondente ai catodi con il connettore J24 della scheda DG02, che contiene i driver a 7 segmenti nella Zona 1 del Laboratorio, raggiungeremo il nostro scopo.

Il cavetto che era già installato univa il connettore J64 (uscite RC0:RC3) della scheda DG06 agli anodi dei diodi LED, come possiamo vedere.

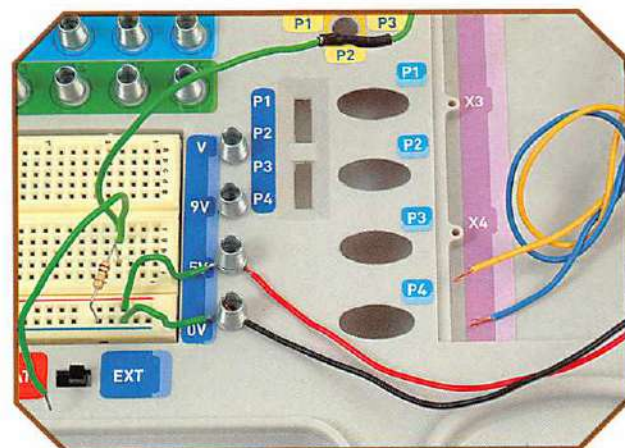
In altre parole, in questo caso i LED rimangono collegati in serie e non si illuminano, perché sono attraversati da una corrente molto bassa, sufficiente per attivare il driver, ma

insufficiente per ottenere una corretta illuminazione degli stessi.

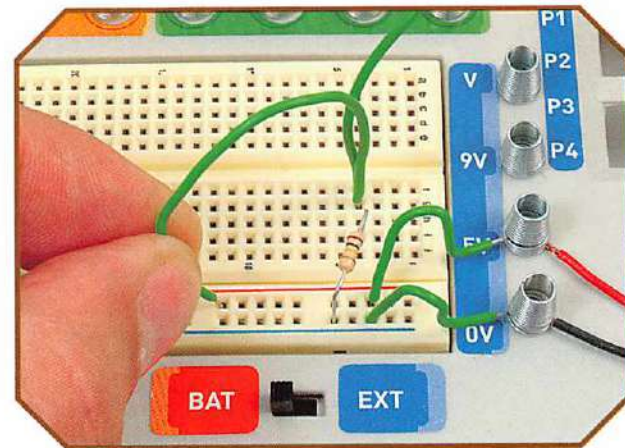
Il montaggio, seguendo le illustrazioni, risulterà rapido e semplice. I quattro terminali del connettore J24 sono collegati direttamente al driver del display 4511, quindi risulta indispensabile mantenere la corrispondenza con i colori dei fili, perché invertendo il collegamento del cavetto la sequenza risulterà al rovescio, e la visualizzazione non si produrrà o avverrà in modo errato. Non si deve utilizzare il connettore J25.

## Ingressi

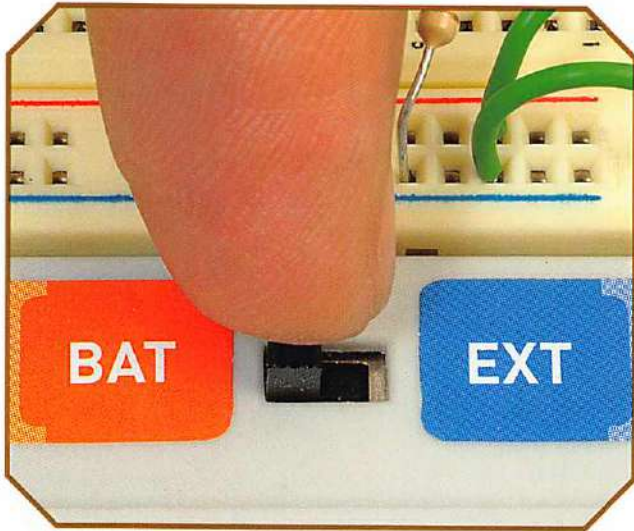
Dato che il funzionamento del PIC è lo stesso di quello dell'esperimento precedente, l'ingresso si implementa nello stesso modo, e si



Collegamenti dell'alimentazione e dell'ingresso di controllo.



Utilizzo dell'ingresso di controllo.

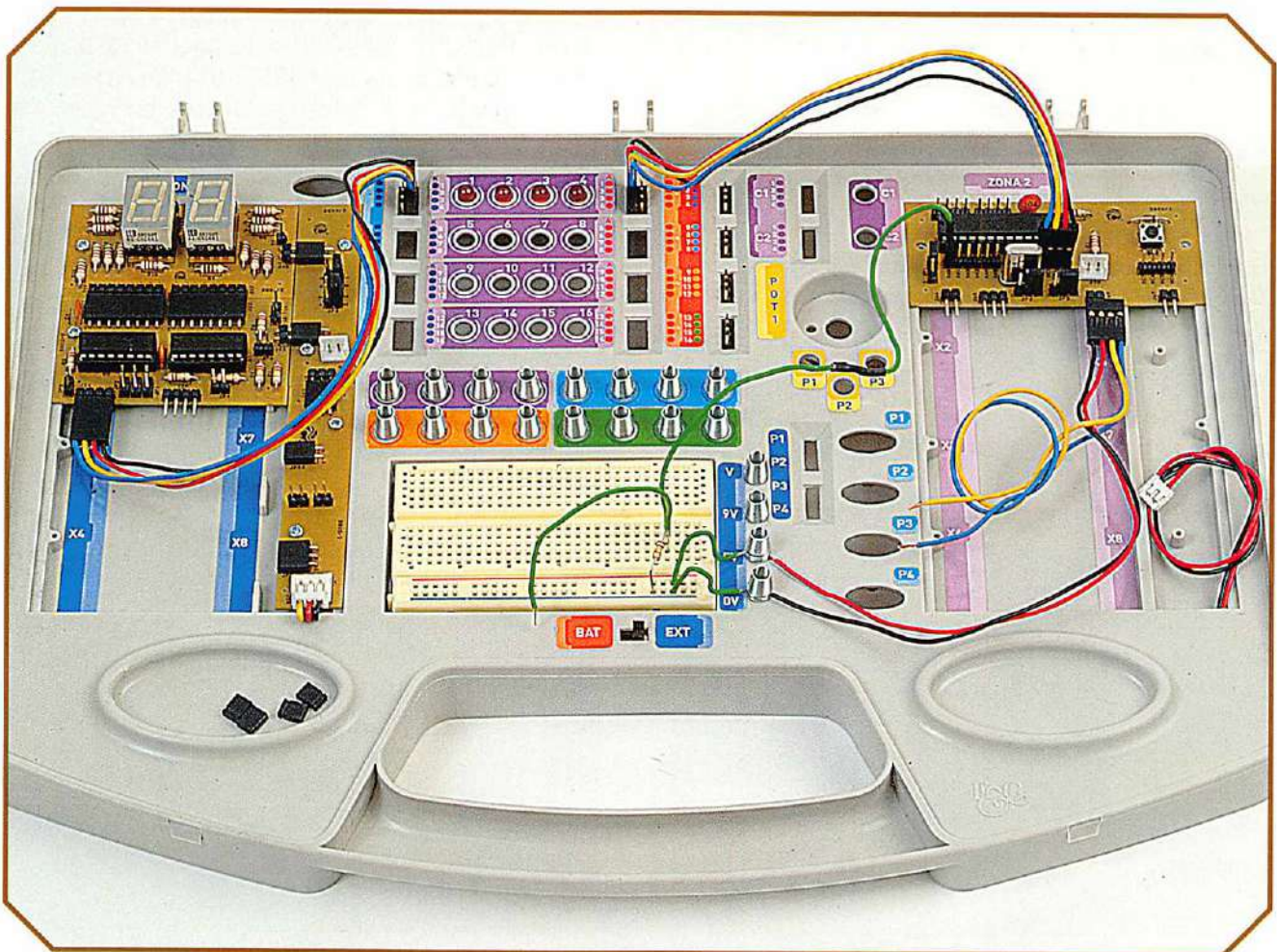


Prima di spostare su BAT  
bisogna verificare tutti i collegamenti.

utilizza il terminale 0 della porta RA e una resistenza da 100 K. Quando l'ingresso RA0 è a livello "1", si visualizzano i numeri in modo sequenziale e quando passa a livello "0", si vede il numero casuale ottenuto.

### Alimentazione

L'alimentazione della scheda DG06 si realizza tramite le due molle 0 e 5 V. Si utilizza un cavetto a quattro fili, dei quali ne useremo solamente due, il rosso per il positivo, collegato a 5 V e il nero per il negativo, collegato a 0 V come GND. Le schede DG01 e DG02 si alimentano attraverso la DG04, e devono essere installati i ponticelli di collegamento fra i terminali 1-2 di JP1 e JP2 della DG04. Infine il commutatore di alimentazione deve essere sulla posizione BAT.



Vista generale del laboratorio con l'esperimento.





## Terzo esperimento con il PIC16F870

**C**ontinueremo con il primo progetto, il cui programma è fornito già scritto nel PIC, e di cui conosciamo già la struttura.

*L'illuminazione dei segmenti del display è controllata direttamente dal PIC, senza la necessità di utilizzare il driver.*

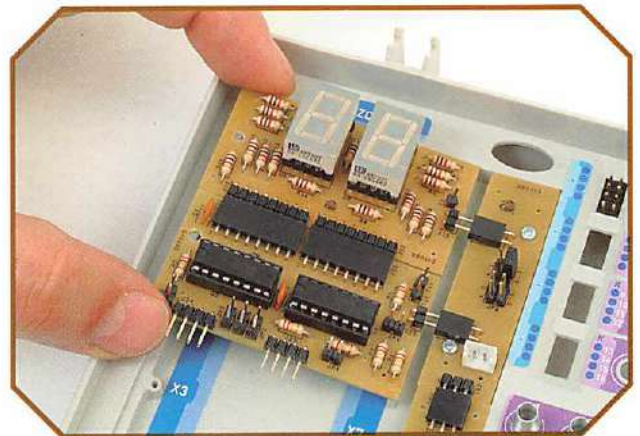
Utilizzeremo l'altra parte del programma, impiegando in questo caso come uscite i terminali da RB0 a RB7 della porta B, e il terminale RA0 della porta A per il controllo.

### Collegamento diretto al display

Un altro modo di risolvere la visualizzazione dei numeri in un display a 7 segmenti, consiste nel collegare quest'ultimo direttamente alle uscite del PIC, senza ricorrere all'utilizzo dei driver.

Conosciamo già la potenza del PIC, quindi ci domandiamo se il microcontroller può realizzare la funzione del driver e fornire le uscite direttamente in codice a 7 segmenti. Con il PIC 16F870 possiamo sostituire il driver e rappresentare un numero in codice a 7 segmenti sul display. L'unica cosa di cui abbiamo bisogno è una porta di uscita libera e creare il codice in assembler che realizzi la conversione.

Per quanto riguarda il codice non è molto difficile creare una subroutine che risolva questa conversione, e in merito alla porta di uscita abbiamo disponibile la Porta B, quindi la configureremo nel programma di uscita e la utilizzeremo per collegare direttamente il display.



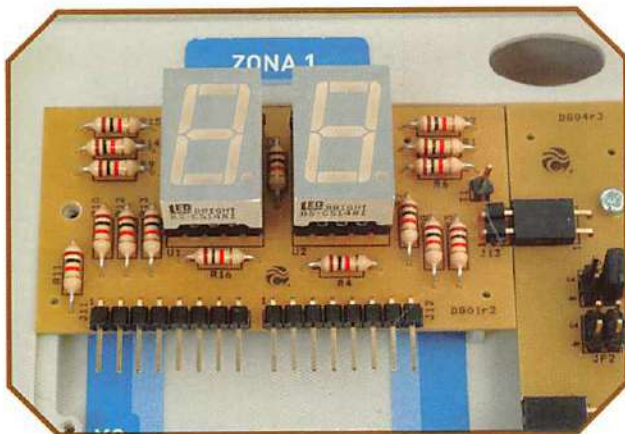
*Bisogna estrarre la scheda DG02.*

È importante notare che il programma contenuto nel microcontroller contempla questa opzione e fornisce sulle uscite della Porta B il numero da rappresentare in codice a 7 segmenti.

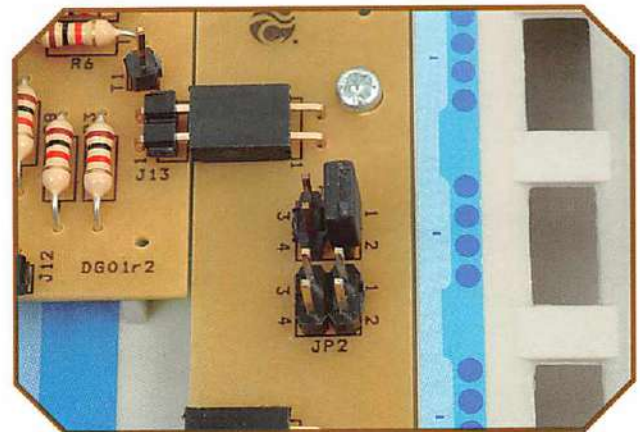
### Il montaggio

Il montaggio di questo esercizio si può dedurre praticamente dalle illustrazioni, però bisogna eseguire alcune preparazioni preventive.

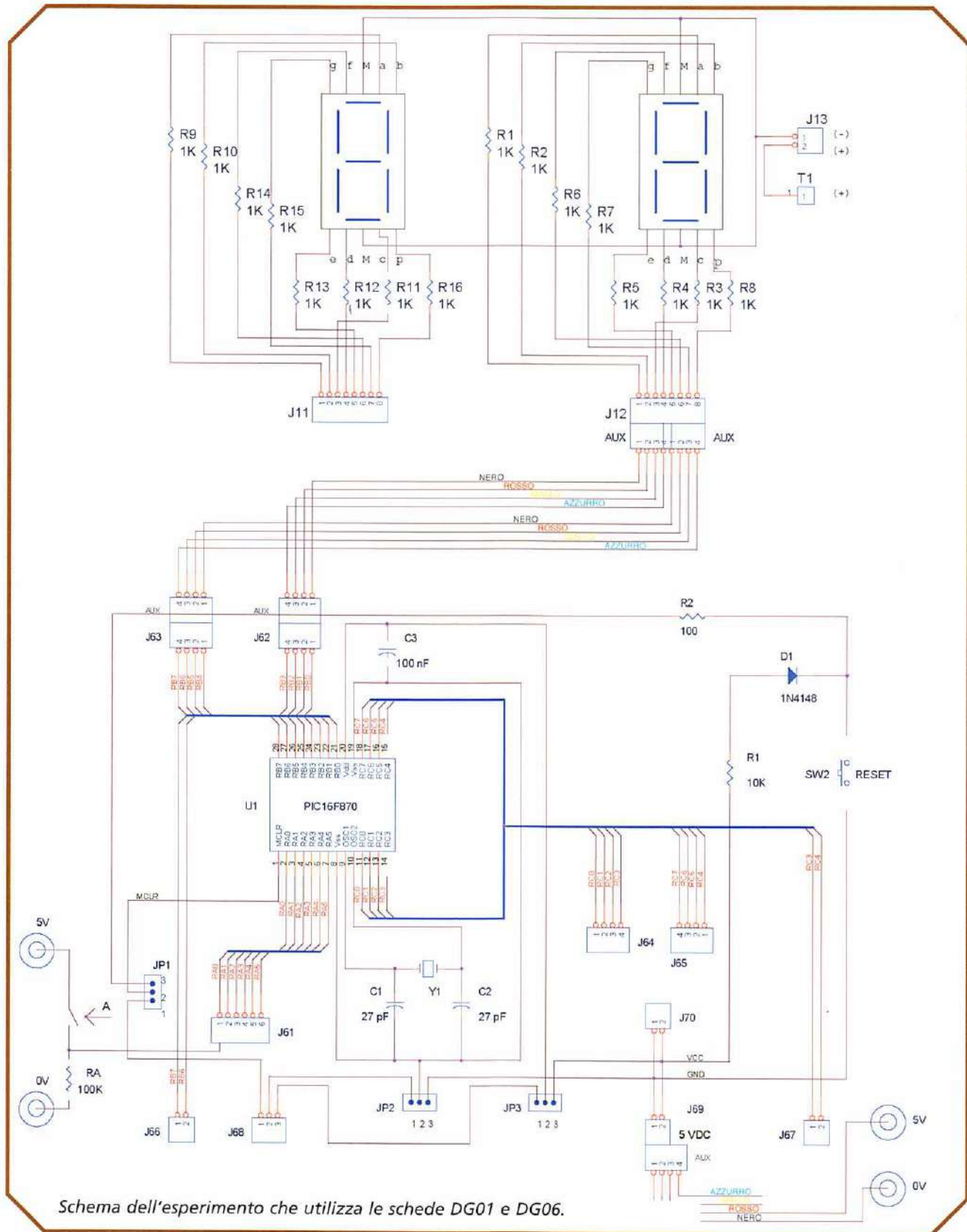
Lasciamo libero l'accesso ai connettori del-



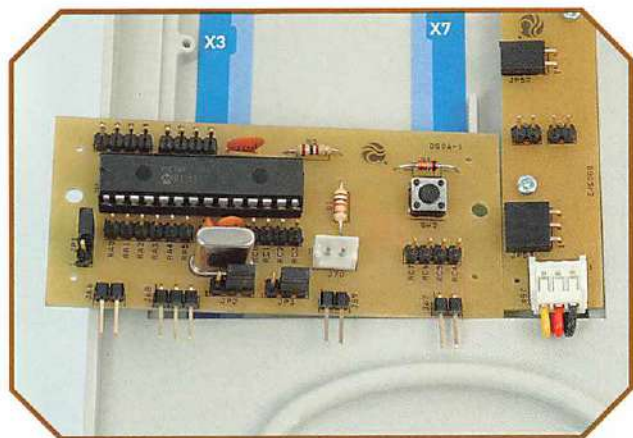
*Scheda DG01  
con i connettori accessibili.*



*La posizione dei ponticelli  
della scheda DG04 in questo esercizio è indifferente.*



Schema dell'esperimento che utilizza le schede DG01 e DG06.



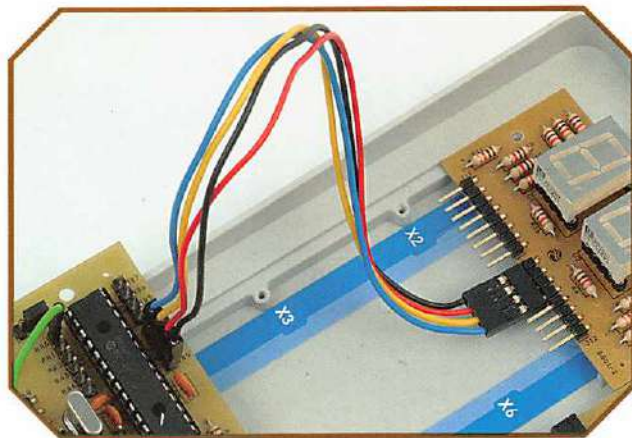
La Scheda DG06  
si deve posizionare vicino alla DG01.

la scheda DG01, togliendo le viti che fissano le schede DG01 e DG02 e staccando entrambe dalla scheda di alimentazione DG04. Dopo aver scollegato le schede DG01 e DG02 dalla DG04, le separeremo, riporremo la scheda DG02 e inseriremo il connettore J13 di DG01 sul connettore J42 di DG04.

Continueremo togliendo dal suo alloggiamento la scheda DG06, che ricordiamo non è ancora stata avvitata.

I terminali della porta B si trovano sulla parte superiore della scheda DG06 che contiene il microcontroller e sono raggruppati di quattro in quattro. In questo modo si facilita il collegamento dei due cavetti a quattro fili.

Con uno dei cavetti si unisce il connettore J62 ai terminali da 1 a 4 in J12 della scheda DG01, tenendo presente l'ordine seguito dai



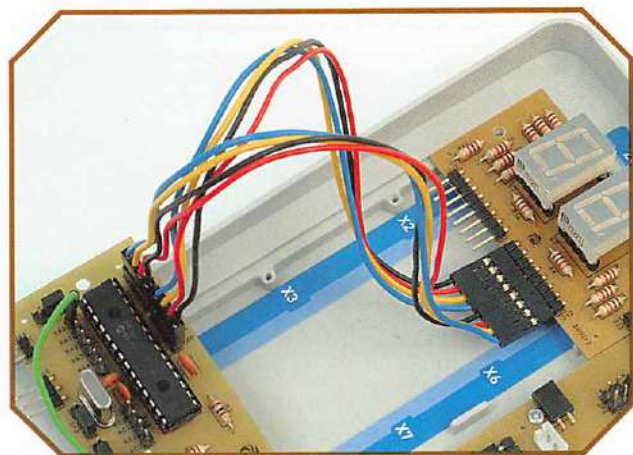
Collegamento del primo  
cavetto fra la porta B e la scheda DG01.

colori. L'altro cavetto unisce il connettore J63 di DG06 ai terminali dal 5 all'8 del connettore J12.

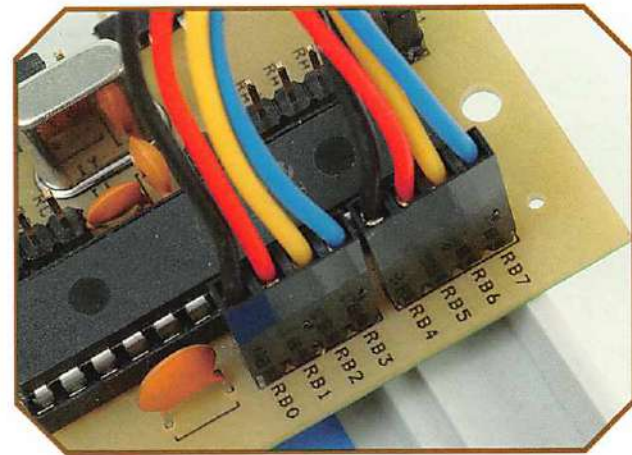
È molto importante non incrociare i cavetti e collegarli in modo adeguato, altrimenti la rappresentazione sul display non sarà quella desiderata e appariranno caratteri strani senza alcun significato. Potete provare a incrociare i cavetti e provare così la visualizzazione risultante.

È necessario inoltre far attenzione a non appoggiare la scheda DG06 sopra zone che potrebbero provocare un cortocircuito, evitando quindi di posarla su oggetti metallici.

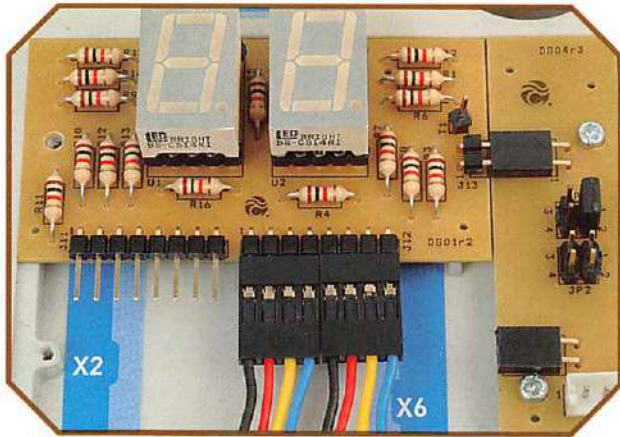
Il connettore J12 si utilizza per i collegamenti dei cavetti. Lo stesso risultato si otterrebbe utilizzando il connettore J11 e l'altro display.



Collegamento del secondo  
cavetto fra la porta B e la scheda DG01.



Collegamento alla porta B,  
bisogna fare attenzione all'ordine dei colori.



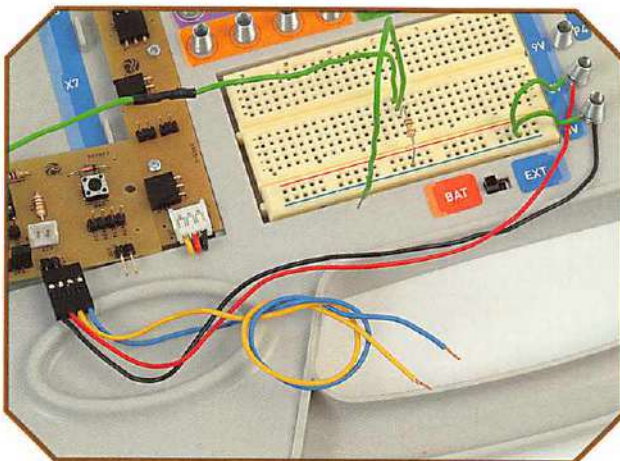
Collegamento a uno dei display.

## Ingresso di controllo

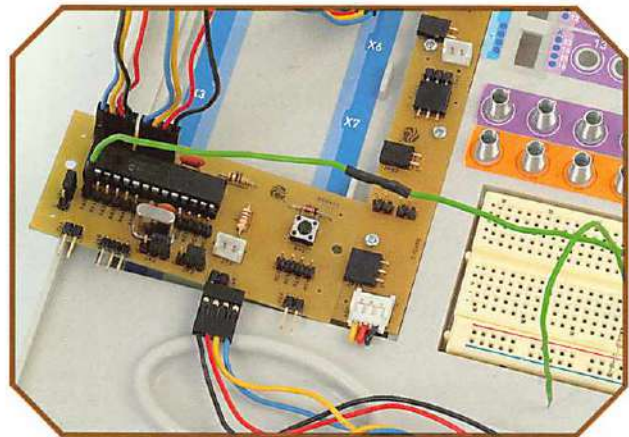
In questo esperimento si utilizza lo stesso ingresso di controllo dei precedenti, il terminale 0 della porta RA e una resistenza da 100 K. Quando l'ingresso RA0 è a livello "1", si visualizzano i numeri in modo sequenziale e passando a livello "0" viene visualizzato il numero casuale ottenuto.

## Alimentazione

L'alimentazione della scheda DG06 si prende dalle molle 0 V e 5 V. Si utilizza un cavetto a quattro fili, dei quali ne impiegheremo solamente due, il rosso per il positivo collegato a 5 V e il nero per il negativo collegato a 0 V come GND. La scheda DG01 si alimenta tramite la DG04 e prende da questa solamente il ne-



L'alimentazione di DG06 è di 5 V.



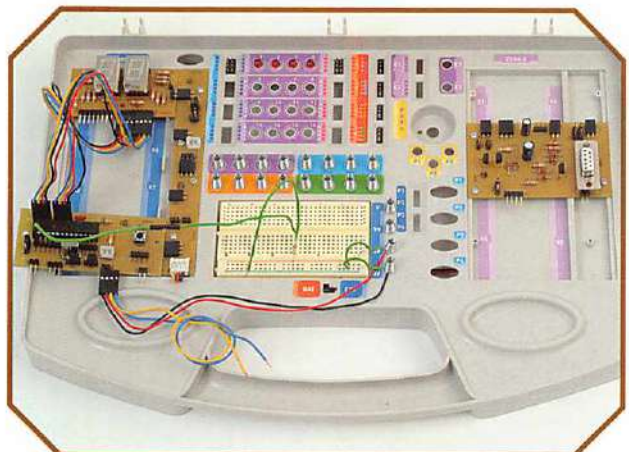
Dettaglio del circuito di controllo, collegato alla porta A, terminale RA0.

gativo, quindi è indifferente che siano collocati o meno i ponticelli della scheda DG04. Infine il commutatore di alimentazione deve essere sulla posizione BAT.

## Conclusioni

Mediante questo esercizio si verifica la versatilità e la flessibilità del laboratorio e del microcontroller. Esistono molti modi per risolvere un progetto, anche se spesso le limitazioni economiche ci indirizzeranno verso un percorso determinato.

Giocate con questo montaggio, ma facendo sempre molta attenzione, in quanto anche se si lavora con tensioni o con correnti molto basse che non possono normalmente causare danni personali, è possibile danneggiare i circuiti.



Laboratorio con l'esperimento realizzato.



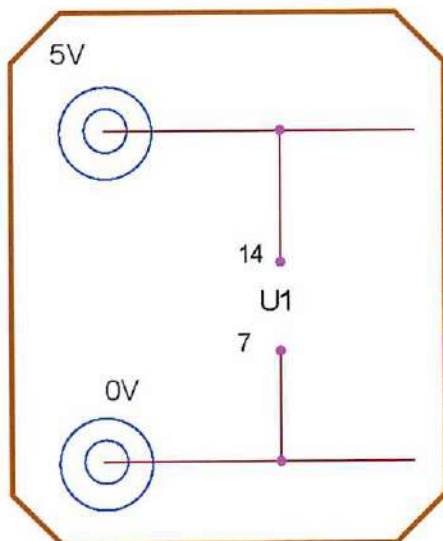
# Flip-flop RS asincrono con porte NOR

**A**nche se il funzionamento sembra lo stesso, i flip-flop RS con porte NOR sono abbastanza differenti da quelli realizzati con le porte NAND.

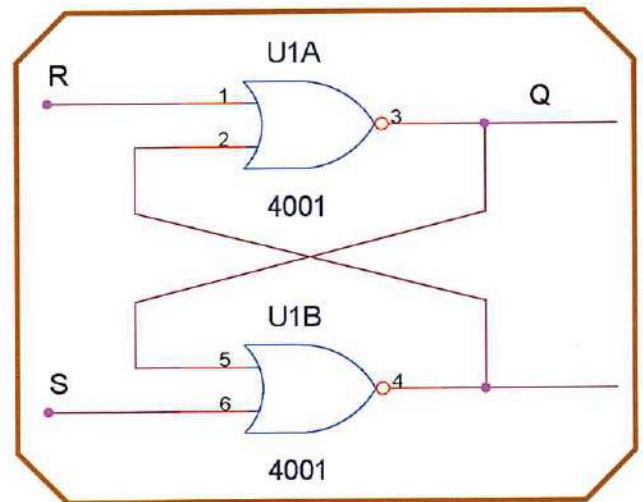
## Il circuito

Il flip-flop RS NOR si riconosce facilmente, perché lo schema è formato dalle porte NOR del circuito integrato U1, che si possono identificare come U1A e U1B.

Adoperiamo un numero di elementi minimo per concentrarci sul funzionamento del flip-flop, utilizzando i collegamenti che hanno come sigle R e S per gli ingressi del flip-flop. All'interno dello schema la prima cosa che richiama la nostra attenzione è che i due ingressi sono collegati al negativo dell'alimentazione con delle resistenze di pull-down, cioè se R e S non sono collegati, entrambi gli ingressi sono a zero. In un flip-flop RS NOR non si devono collegare i due ingressi contemporaneamente a 1, perché in questo caso l'uscita sarebbe indeterminata e il costruttore non garantisce il risultato, in altre parole lo stato stesso delle uscite. Ricordate che in un flip-flop realizzato con porte NAND non si possono collegare simultaneamente a zero, perché



Rappresentazione dell'alimentazione di U1.



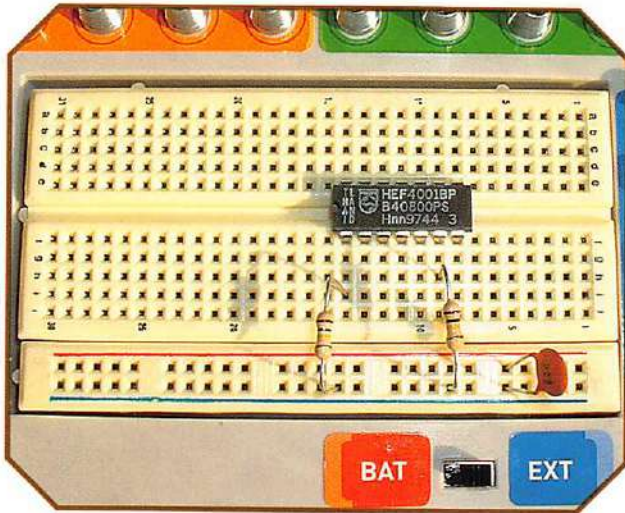
Flip-flop RS con porte NOR.

anche in questo caso l'uscita sarebbe indeterminata.

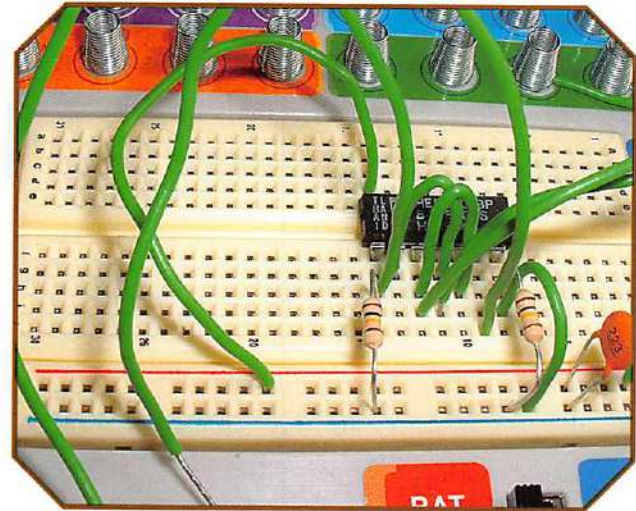
Un'altra differenza è che l'ingresso di SET si trova sulla porta contraria a quella dell'uscita, per questo motivo alcuni autori chiamano il flip-flop RS NOR come SR, invertendo l'ordine delle due lettere.

Se chiudiamo il commutatore S mantenendo aperto R, l'uscita Q che corrisponde al terminale 3 del circuito integrato 4001 passa a livello alto, e rimane in questo stato anche se si riapre S, fino a quando R rimane aperto, in questa condizione il LED 4 della matrice dei LED del laboratorio si manterrà illuminato, e il LED 3 dovrà rimanere spento.

Se con il circuito nello stato precedente colleghiamo nuovamente S, l'uscita non cambia ma se una volta aperto S colleghiamo R, cambia lo stato del circuito attivando l'uscita negata, quindi il terminale 4 dell'integrato passa a 1, mentre il 3 passa a 0; si spegne quindi il LED 4 e si illumina il LED 3, quest'ultimo rimarrà acceso anche se si apre il collegamento R.



Componenti sulla scheda Bread Board.

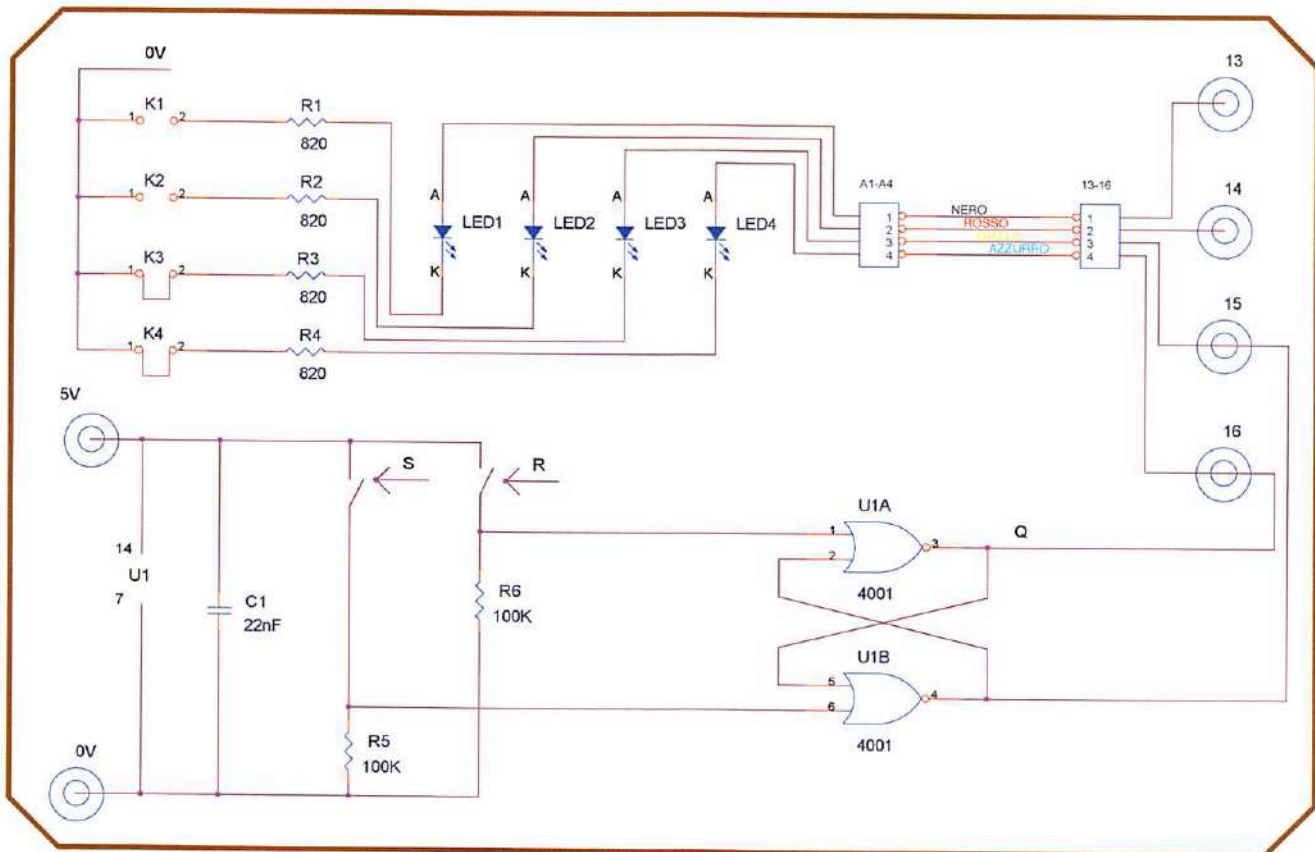


Cablaggio interno.

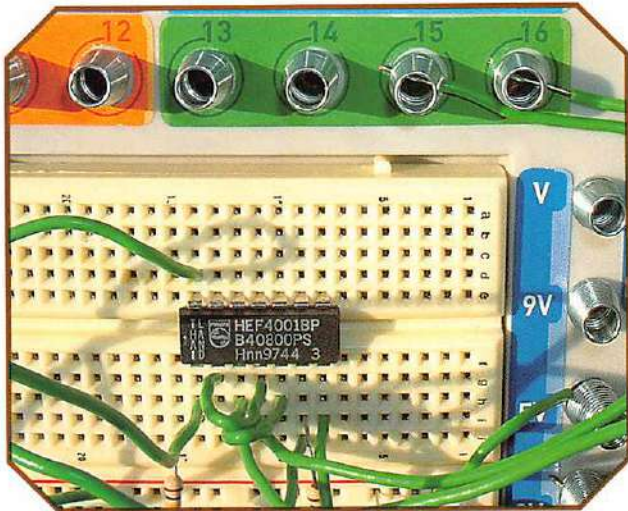
Con i due commutatori il LED 3 deve continuare a essere illuminato.

Riassumendo, collegando momentaneamente il collegamento 5 si attiva l'uscita di U1A, terminale 3, e si illumina il LED 4, mentre

attivando R si attiva l'uscita della porta U1B e si illumina il LED 3, senza dimenticare che R e S non si possono attivare contemporaneamente dato che in questo caso non sarà garantito lo stato dell'uscita.



Flip-flop RS asincrono con porte NOR.



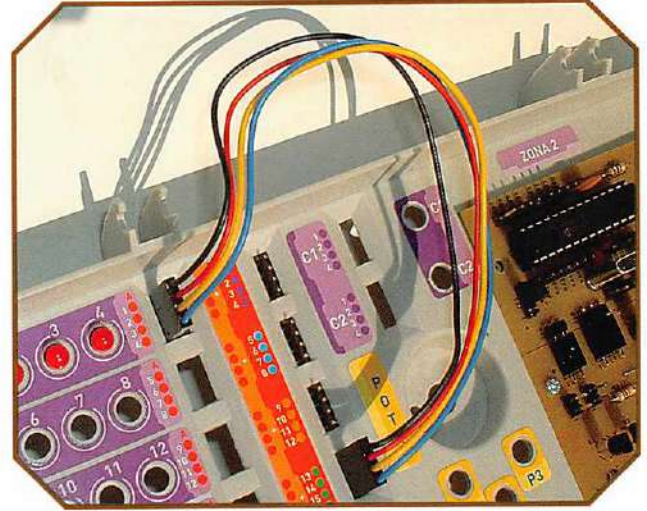
Collegamenti alle molle 15 e 16.

### Montaggio

La realizzazione pratica di questo esperimento è piuttosto veloce e semplice, in quanto nonostante lo schema sia grande, molti dei componenti sono permanentemente montati sul laboratorio, come ad esempio i quattro LED e le loro corrispondenti resistenze di polarizzazione.

È sufficiente seguire con attenzione lo schema, inserire i componenti sulla scheda Bread Board e realizzare il cablaggio con attenzione. In questo caso l'unica precauzione particolare è l'orientamento del circuito integrato.

Bisogna collegare gli anodi dei LED alle molle dalla 13 alla 16, per questo utilizzare-



Collegamenti con il cavetto fra A1-A4 e 13-16.

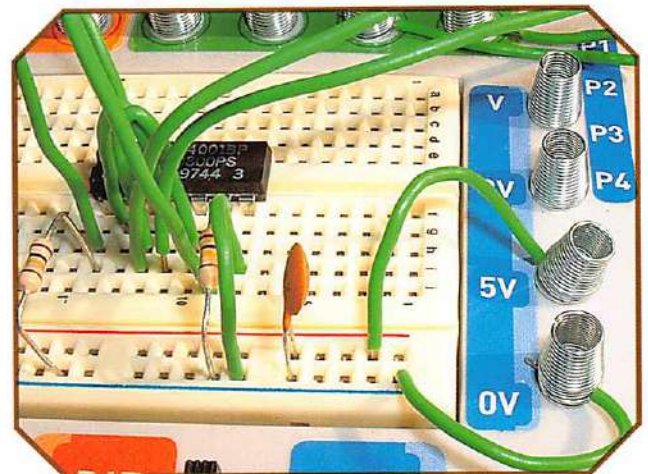
mo un cavetto a 4 fili, terminato su connettori a quattro vie per stabilire la connessione fra i terminali degli anodi dei LED A1, A2, A3, A4 e i terminali dal 13 al 16 di accesso alle molle.

### Alimentazione

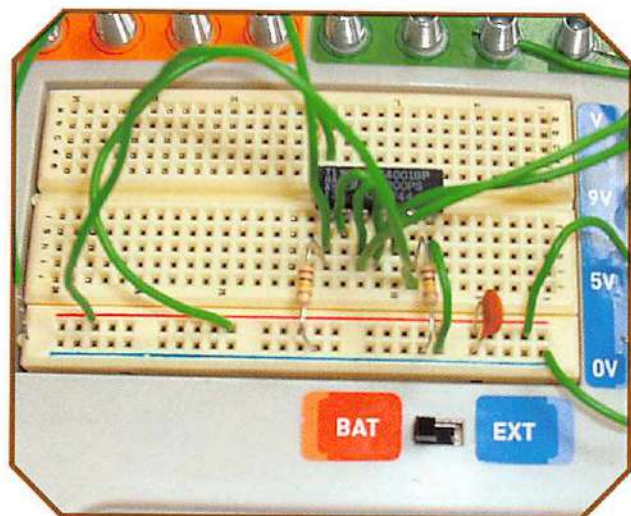
Questo esperimento si alimenta con 5 V, o con i 4,5 delle pile; il positivo e il negativo dell'alimentazione si prendono direttamente dalle molle identificate come 0 e 5 V, senza dimenticare di realizzare il cablaggio della Bread Board per l'alimentazione del circuito integrato: il terminale 7 si collegherà a 0 V e il terminale 14 a 5 V. Nello schema è stata utilizzata una nuo-



I ponticelli devono essere collocati su K3 e K4 per collegare l'alimentazione ai catodi dei LED.



Collegamenti alle molle di alimentazione.

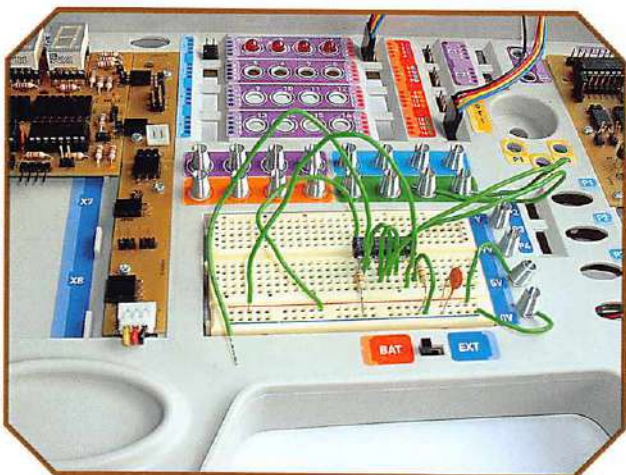


Il commutatore deve essere sulla posizione BAT.

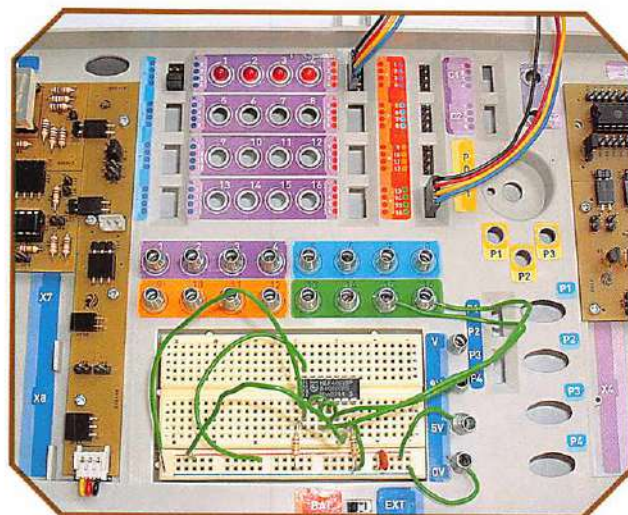
va rappresentazione per l'alimentazione dell'integrato, che è quella normale nei progetti complessi ottenendo così l'eliminazione di alcune linee di schema per una maggiore chiarezza dello stesso. È assolutamente necessario collocare i ponticelli di alimentazione su K3 e K4 per alimentare il negativo dei LED. Il condensatore C1 è un filtro di alimentazione e di uso comune in questo tipo di circuiti.

## Prova

Prima di collegare l'alimentazione bisogna rivedere tutto il lavoro svolto, e se risulta corretto andremo a verificare lo stato delle pile nel portabatterie situato sotto la zona 1, quindi



Collegando R si illumina il LED 15.



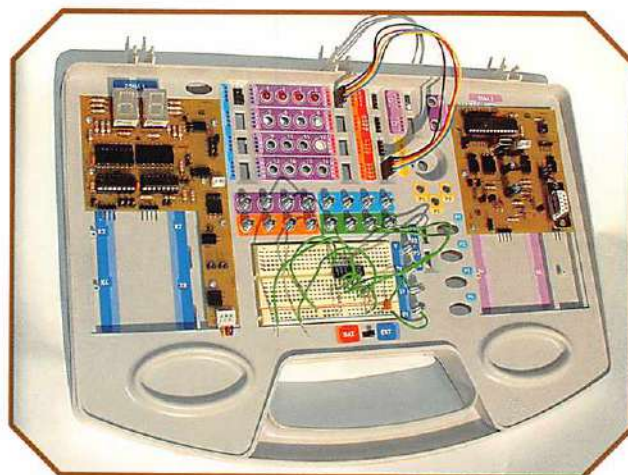
Collegando S si illumina il LED 16.

passeremo il commutatore sulla posizione BAT.

I collegamenti R e S devono essere aperti. Nel momento della connessione si può illuminare qualunque dei due LED, però se colleghiamo S si deve accendere il LED 4, che rimarrà illuminato anche aprendo il collegamento; se si collega R si deve illuminare il LED 3 e deve rimanere così anche scollegando nuovamente R.

## LISTA DEI COMPONENTI

U1	Circuito integrato 4001
R5, R6	Resistenza 100 K (marrone, nero, giallo)
C1	Condensatore 22 nF



Vista del laboratorio con l'esperimento.





# Registri di spostamento

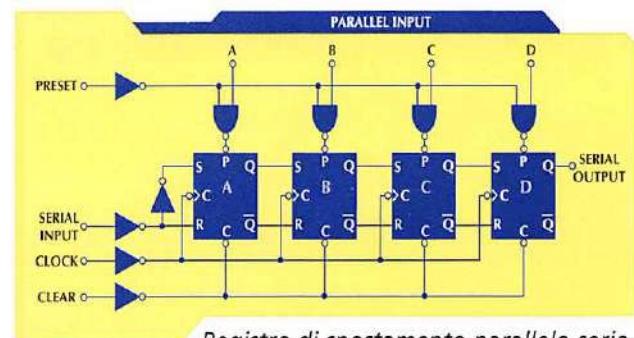
**U**n registro di spostamento è composto da un catena di flip-flop collegati in cascata, cioè l'uscita di uno è collegata all'ingresso dell'altro. Nella nomenclatura inglese si chiamano **SHIFT REGISTER**.

## Registro di spostamento

I flip-flop sono celle elementari di memoria, e ognuna di esse contiene un bit, quindi se un registro di spostamento è composto da 8 flip-flop collegati in cascata si dice che il registro può contenere 8 bit. La denominazione "spostamento" è dovuta al fatto che l'informazione si sposta attraversando in sequenza tutti i flip-flop seguendo il ritmo segnato dalla frequenza del clock; in alcuni casi questo segnale è chiamato segnale di spostamento, infatti a ogni impulso di clock l'informazione passa da un bistabile al successivo.

L'ingresso del clock è comune a tutti i flip-flop, l'ingresso è sincrono R-S o J-K. Normalmente nel caso R-S si collega solamente la S dalla quale si deriva la R tramite un inverter per evitare le indeterminazioni sull'ingresso; questo ingresso normalmente riceve il nome di ingresso serie, all'uscita dell'ultimo flip-flop si ottiene lo stesso segnale applicato all'ingresso, però ritardato di tanti impulsi di clock quanti sono i flip-flop che compongono il registro.

È possibile disporre anche di un collegamento all'uscita di ogni flip-flop, in questo modo si ottiene un'uscita parallela, nel caso specifico però è necessario conoscere il momento in cui l'ingresso serie è completo, altrimenti se il segnale in parallelo viene preso nel momento sbagliato, ad esempio un impulso



Registro di spostamento parallelo-serie.

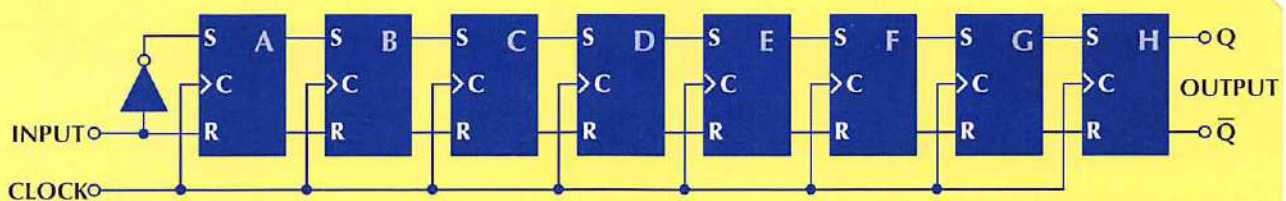
prima o un impulso dopo, l'informazione ottenuta sarà errata.

## Ingressi asincroni

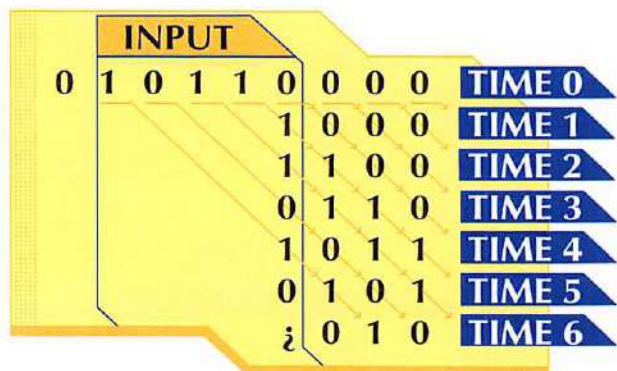
Alcuni registri di spostamento hanno dei bistabili con ingressi asincroni, cioè Preset e Clear, che permettono di far arrivare al registro informazioni in parallelo.

## Ingresso parallelo, uscita serie

Fra i molti circuiti possibili, ne abbiamo scelto uno fra i più classici che si trova frequentemente nella letteratura di settore. Osserviamo il suo schema per rendere più facile la spiegazione: ha quattro flip-flop collegati in cascata e un ingresso serie, SERIAL INPUT; i due inverter evitano le indeterminazioni sull'ingresso, se utilizziamo come ingresso "SERIAL INPUT"



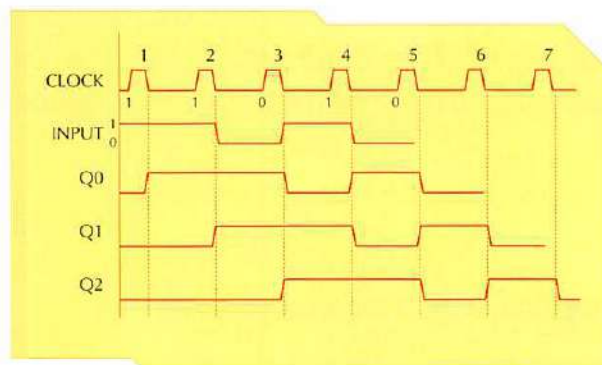
Registro di spostamento serie-serie.



Evoluzione di una sequenza di bit con gli impulsi di clock.

torniamo nel caso precedente di un registro con ingresso e uscita serie, però se osserviamo lo schema vedremo nella parte superiore dello stesso quattro porte NAND che si possono utilizzare per impostare l'uscita di ogni flip-flop a 1; prima di fare questo però bisogna eseguire un'operazione di cancellazione con CLEAR, applicando un 1 a questo ingresso, in modo che tutte le uscite passino a 0, dopodiché con il preset attivo a livello basso, il livello di ogni ingresso (A, B, C e D) viene portato sull'uscita di ogni bistabile, successivamente verrà portato sull'uscita serie in modo sequenziale a ogni impulso di clock, fino a completare tutta la sequenza.

È chiaro che si richiede un circuito esterno



Segnali della frequenza precedente, in un diagramma dei tempi.

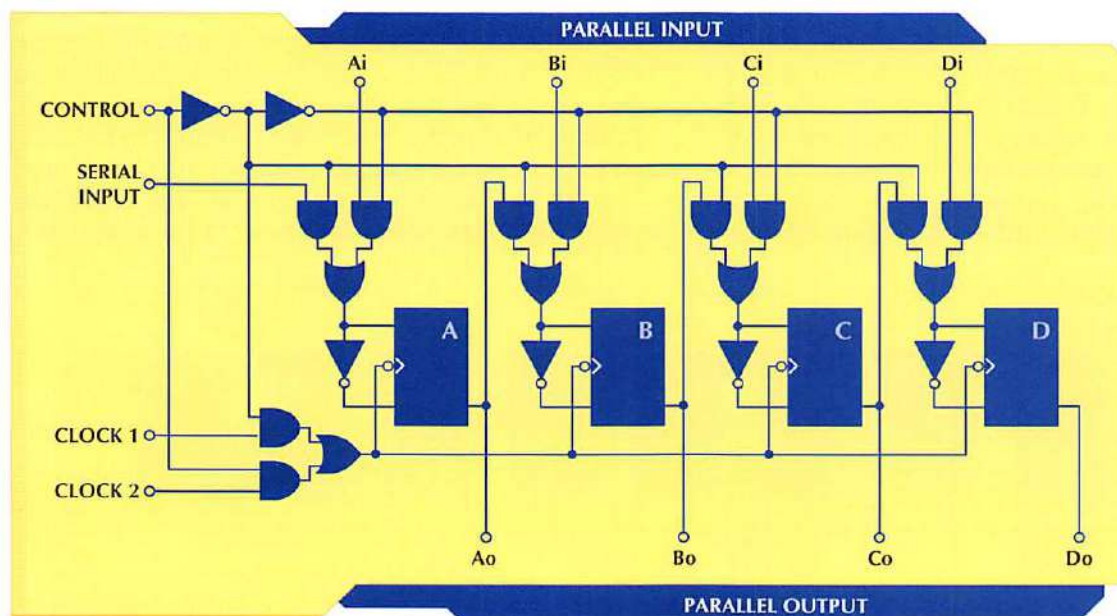
addizionale che abbia il compito di determinare quando la sequenza di uscita è completa.

L'utilizzo di queste porte NAND, permette applicando un 1 a questi ingressi, di ignorare l'informazione applicata agli ingressi paralleli (A, B, C, e D) rendendo possibile l'utilizzo dell'ingresso serie.

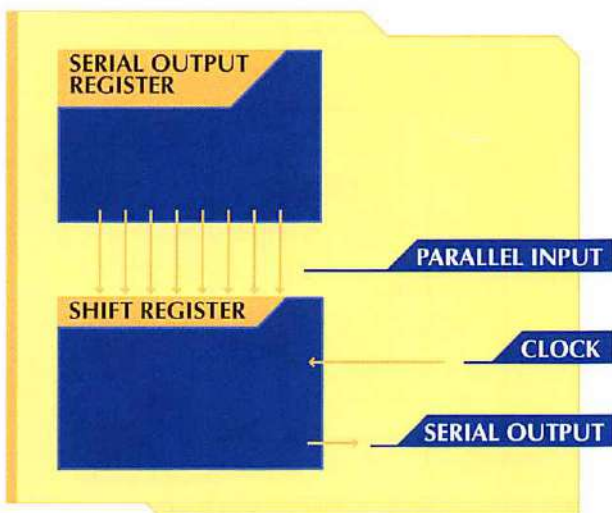
### Circuito S-S, P-P

Vedremo ora un esempio di circuito più completo che può funzionare in modo serie-serie o parallelo-parallelo, esamineremo inoltre come sia possibile spostare il segnale verso destra o verso sinistra.

Se osserviamo il circuito ci renderemo conto



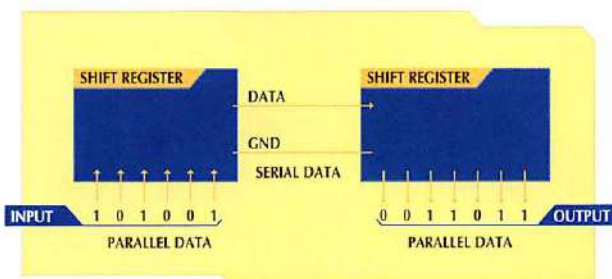
Registro di spostamento che può funzionare in modo serie-serie e in modo parallelo-parallelo.



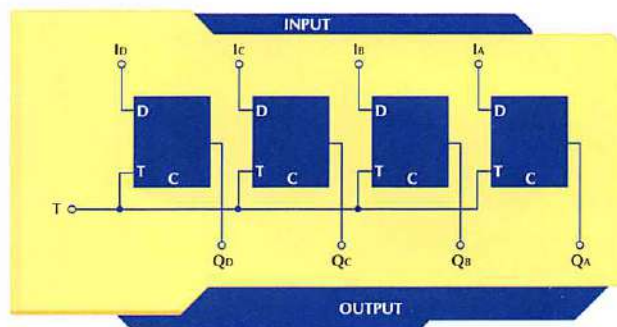
Conversione da parallelo a serie.

che applicando uno 0 all'ingresso di controllo si chiudono le porte AND identificate nello schema come P, ignorando gli ingressi paralleli. Nel caso contrario, cioè se l'ingresso di controllo viene impostato a 1, si chiudono le porte identificate con una S, si taglia la catena di flip-flop e l'uscita di ognuno di essi si scollega dall'ingresso del successivo.

Per fare in modo che il segnale si sposti in un verso o nell'altro, è necessario collegare ogni uscita a ogni ingresso nel modo adeguato. Ad esempio per lo spostamento verso sinistra, l'ingresso è quello dell'ultimo bistabile - ingresso D - l'uscita di quest'ultimo si porta sull'ingresso C, la cui uscita deve essere portata sull'ingresso B e l'uscita di B all'ingresso di A, essendo l'uscita di A l'uscita del registro con spostamento verso sinistra. Affinché tutto questo funzioni bisogna applicare un 1 all'ingresso di controllo, dato che in caso contrario i collegamenti nell'altro senso sono già fatti.



Una delle applicazioni dei registri di spostamento è nella trasmissione dei segnali.



Registri di trasferimento.

### Registri fissi

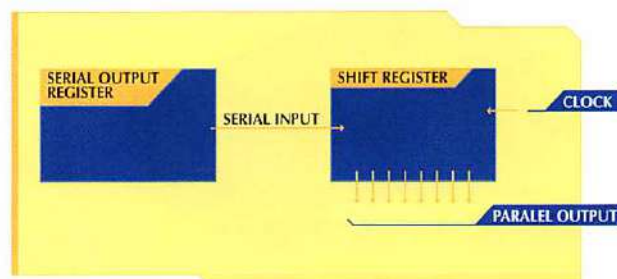
Questi registri possono mantenere l'informazione per un tempo quasi indefinito, però esistono altri tipi di registri che non utilizzano i flip-flop, sono molto più veloci, ma possono mantenere l'informazione per minor tempo, anche se sufficiente per alcune applicazioni, questo per ricordare che non tutti i registri di questo tipo sono uguali.

### Registri di trasferimento

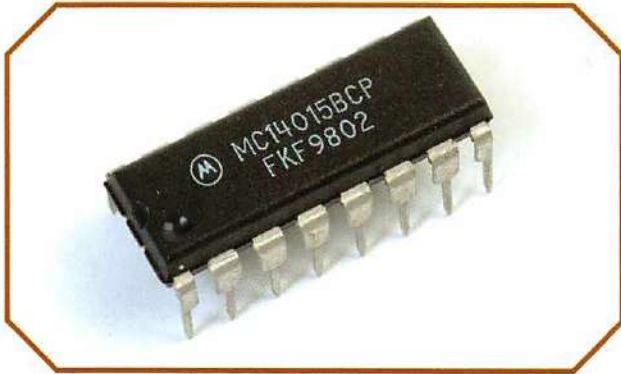
In questo tipo di registri l'informazione entra ed esce in parallelo, si utilizzano come memorie temporali, sono formati da flip-flop tipo D e hanno un ingresso di controllo comune T, per cui l'informazione esce dal registro così com'è entrata.

### Conversione da parallelo a serie

Una delle applicazioni dei registri di spostamento è la trasmissione dei segnali, dato che normalmente si elaborano in parallelo ma si trasmettono in serie, quindi è sufficiente una



Conversione da serie a parallelo.



Il 4015 ha due registri da 4 bit, con ingressi serie e uscite parallele.

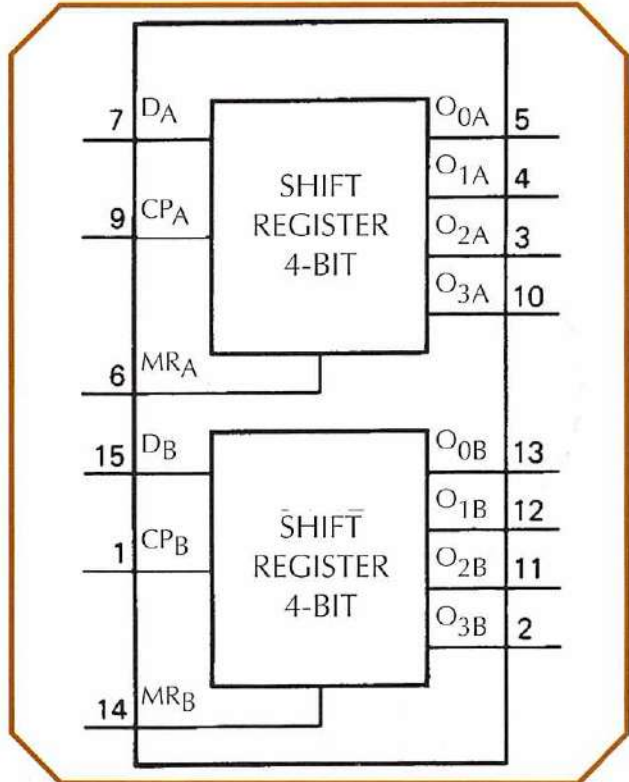
conversione da parallelo a serie. L'uscita di un registro che contiene l'informazione in parallelo si porta sull'ingresso parallelo di un registro di spostamento, e si ottiene sulla sua uscita un segnale serie che si può trasmettere facilmente, il controllo è realizzato da un clock che genera il segnale di spostamento.

### Conversione da serie a parallelo

Il segnale serie si converte in parallelo utilizzando anche qui un registro di spostamento. In questo caso si utilizza un ingresso serie e una uscita in parallelo ed è evidente che è necessario sincronizzare tutti i circuiti, per poter recuperare il segnale originale, però questo è già un altro argomento.

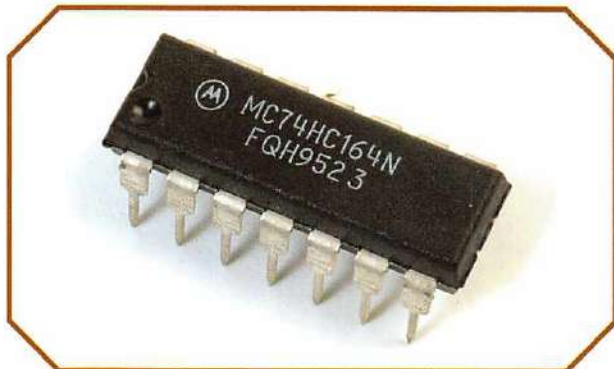
### Circuiti commerciali

Sono facilmente reperibili sul mercato molti circuiti integrati che contengono registri di spostamento, è quindi necessario consultare il

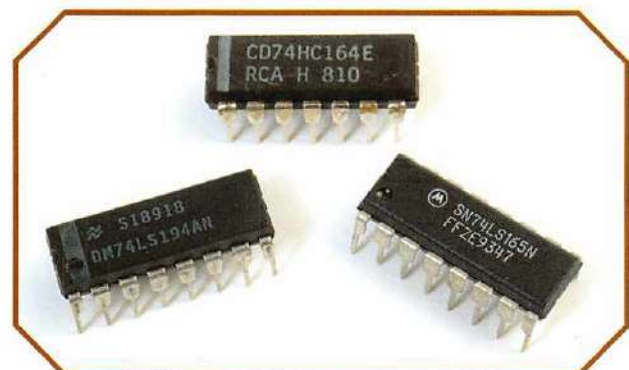


Rappresentazione del 4015.

catalogo di ogni costruttore per conoscere le caratteristiche di ognuno di essi. Di seguito citeremo alcuni esempi: il 4015 della famiglia CMOS contiene 2 registri da quattro bit ciascuno con ingresso serie e uscita parallela, il 74HC164 è da otto bit, con ingresso serie e uscita parallela, al contrario il 74HC165 ha un ingresso parallelo e uscita serie. Il 74HC194 ha un ingresso da quattro bit in parallelo, uscita parallela e possibilità di spostamento a destra e a sinistra e due ingressi serie.



Il 74HC164 contiene un registro di spostamento da 8 bit con ingresso serie e uscite parallele.



Circuiti integrati e registri di spostamento.



# Prova del contatore a due digit

*due contatori che abbiamo a disposizione contano gli impulsi, quindi è evidente che per la loro prova abbiamo bisogno di un circuito che generi degli impulsi.*

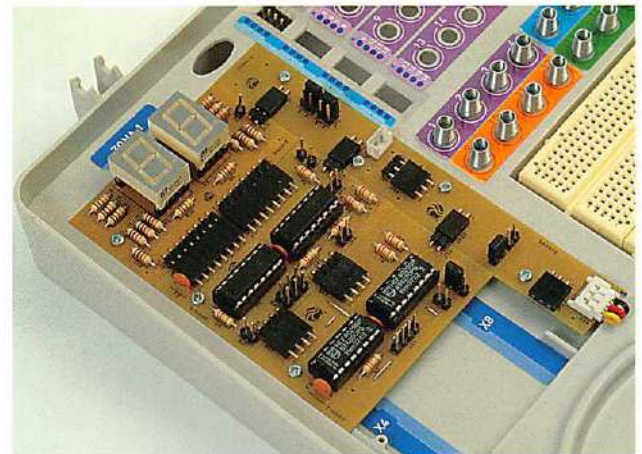
## Il circuito

L'obiettivo di questo esperimento è verificare il funzionamento del contatore a due digit montato nella zona 1 del laboratorio, questo contatore è già installato e rimane montato in modo permanente, questo facilita la sua rapida disponibilità per la realizzazione di esercizi futuri dove sia necessario utilizzare un contatore.

Esistono molti modi di provare un contatore, ma uno dei migliori è l'utilizzo di un generatore d'impulsi, la cui uscita venga utilizzata per eccitare l'ingresso di clock del contatore. Per fare in modo che sia possibile visualizzare il conteggio, la frequenza del clock deve essere molto bassa.

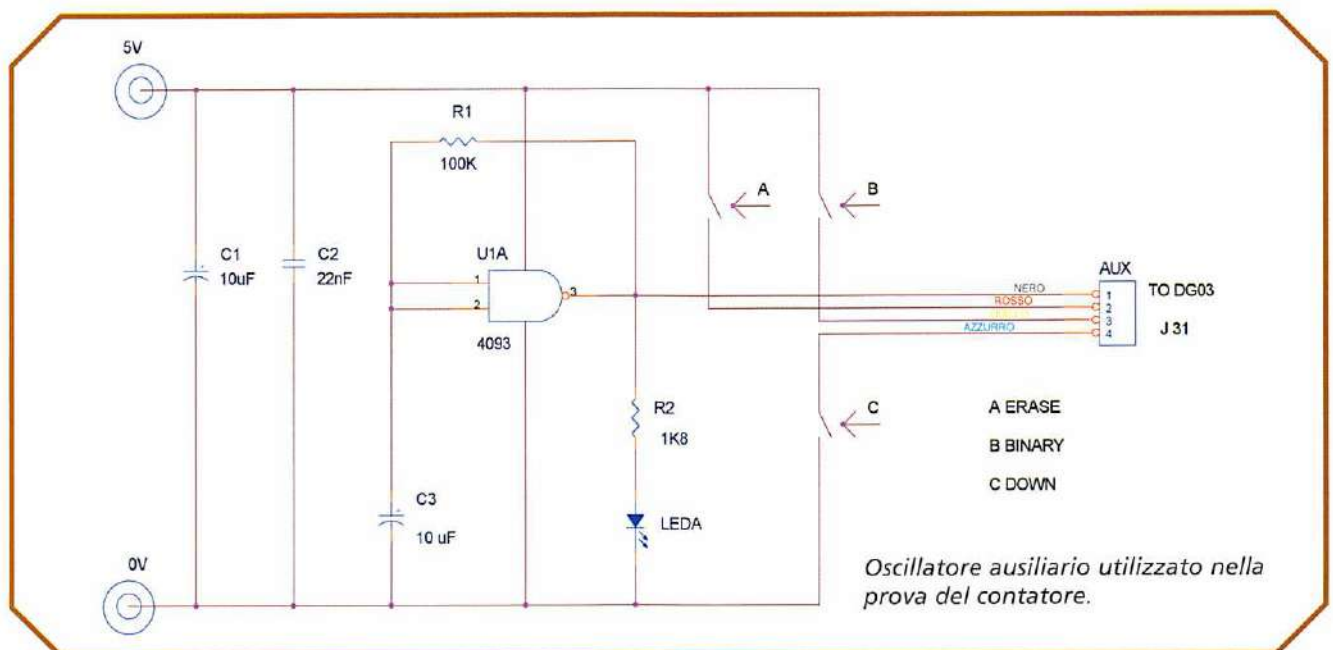
## Il generatore di impulsi

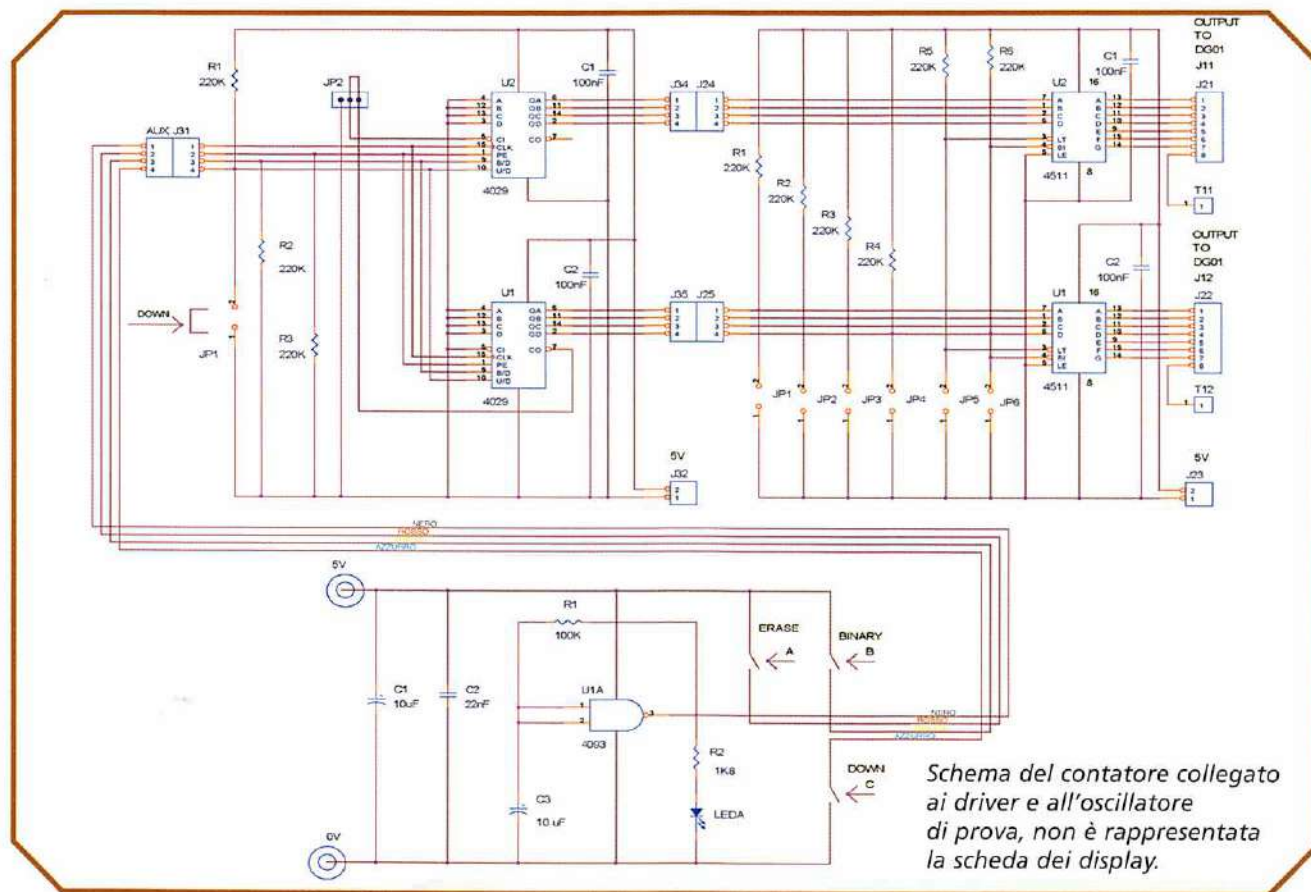
Il generatore di impulsi è molto semplice, utilizza una porta del circuito integrato 4093, una resistenza R1 da 100 K e un condensatore da 10  $\mu$ F, l'uscita si prende dall'uscita della porta



Le schede DG01, DG02 e DG03 formano un contatore a due digit.

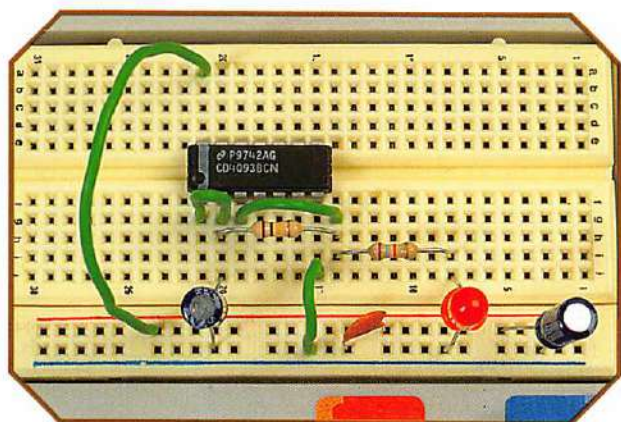
U1A, che corrisponde al terminale 3 del circuito integrato. La resistenza R2 limita la corrente che circola sul LED, e quest'ultimo è utilizzato per verificare che l'oscillatore funzioni. I condensatori C1 e C2 filtrano l'alimentazione.





## Montaggio

Prima di iniziare il montaggio è necessario verificare che l'alimentazione sia scollegata, riguardo a questo vi consigliamo di lasciare per ultimo il collegamento del positivo dell'alimentazione della scheda Bread Board. Per scollegare l'alimentazione del contatore è



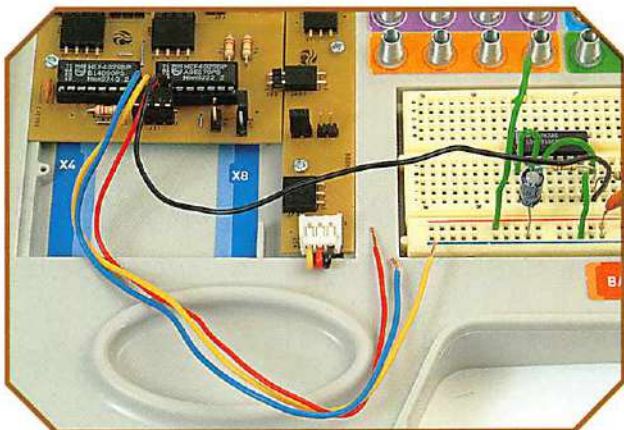
L'oscillatore si monta sulla scheda Bread Board.

sufficiente togliere tutti i ponticelli delle schede DG04 e DG05. Dato che il contatore è già disponibile è sufficiente montare i componenti del generatore degli impulsi sulla scheda Bread Board e realizzare il cablaggio facendo particolare attenzione al montaggio del circuito integrato, alla polarità dei condensatori elettrolitici e a quella del diodo LED.

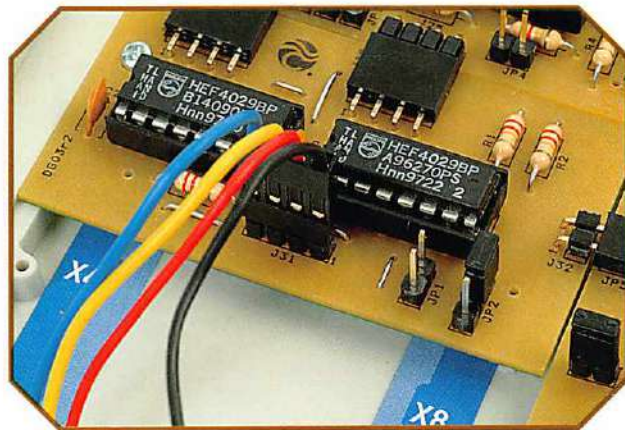
I collegamenti al contatore si realizzano con un cavo a quattro fili, terminato su un connettore nero a quattro vie, che si collega direttamente al connettore J31 della scheda DG03, facendo attenzione a collegare il filo nero sul terminale 1. Non deve essere collegato nessun ponticello sulla scheda DG01 e DG02, sulla scheda DG03 invece deve essere collocato JP2 in posizione normale, si può verificare la corretta posizione dalle fotografie.

## Alimentazione

Sia il generatore di impulsi che il contatore, possono funzionare a 5 V oppure a 9 V, però per fare in modo che i livelli di uscita del generatore di impulsi siano compatibili con gli in-



Dettaglio dei collegamenti con il cavetto fra J31 di DG03 e la scheda Bread Board.



Sul connettore J31, il filo nero si collega al pin numero 1.

gressi del contatore, entrambi i circuiti dovranno essere alimentati alla stessa tensione.

La tensione di alimentazione del contatore è selezionata con i ponticelli delle schede DG04 e DG05. Per 5 V i ponticelli della scheda DG04 devono essere nella posizione 1-2, mentre sulla DG05 il ponticello deve essere collocato su JP1.

Per alimentare a 9 V i ponticelli della scheda DG04 devono essere sulla posizione 3-4, mentre sulla DG05 il ponticello deve essere su JP2.

Per quanto riguarda l'oscillatore si utilizzano le molle di alimentazione da 5 V oppure da 9 V, senza dimenticare di collegare il negativo dell'alimentazione a 0 V.

### Prova

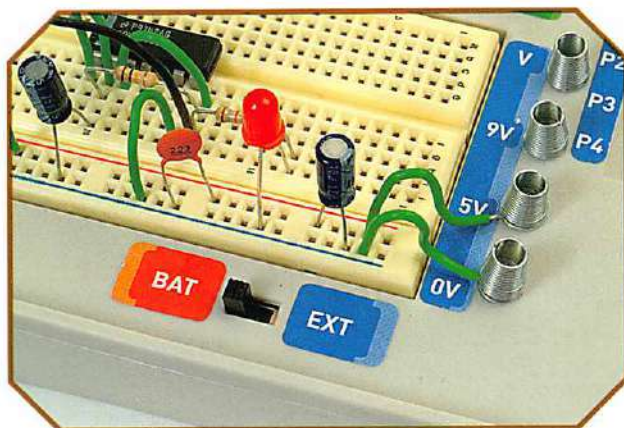
Dopo aver realizzato il montaggio in modo corretto è sufficiente collegare l'alimentazio-

ne, tutti i circuiti alla stessa tensione, e il filo nero del cavetto al terminale 3 del circuito integrato, lasciando i tre fili rimanenti scollegati.

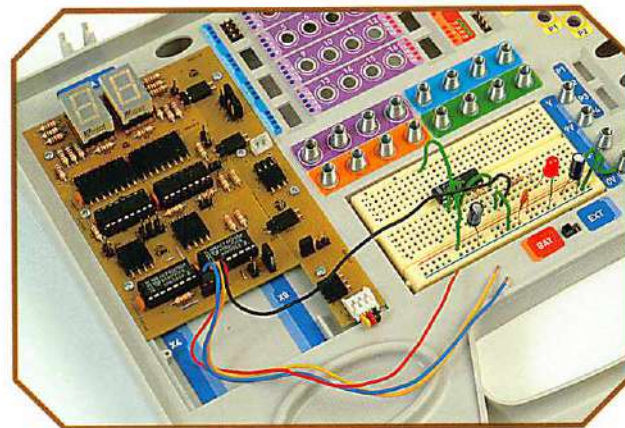
Il contatore deve iniziare il conteggio con i numeri naturali e in avanti. La resistenza del generatore di impulsi R1 si può aumentare di valore per fare in modo che il contatore vada più lentamente, o viceversa.

### Contatore decrescente

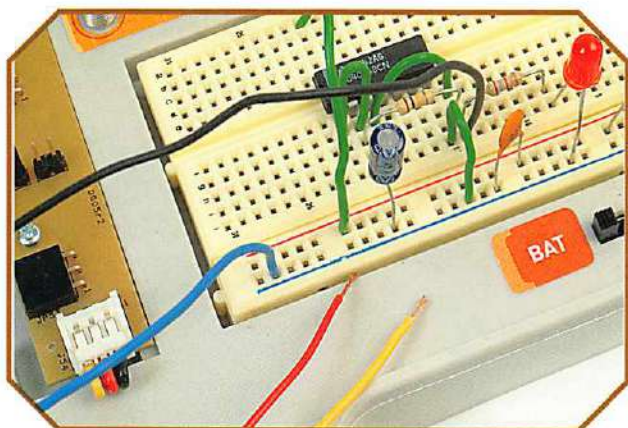
Con il contatore in funzionamento, se eseguiamo il collegamento indicato sullo schema come C, cioè se colleghiamo il negativo dell'alimentazione sul filo azzurro, il contatore conterà in senso discendente, per tutto il tempo in cui questo filo sarà collegato. Possiamo anche eseguire questo collegamento utilizzan-



Collegamento dell'alimentazione a 5 V, commutatore su BAT durante la prova.



A, B e C si possono lasciare volanti, il contatore deve contare in avanti.



Collegando C, il contatore conta in senso decrescente.

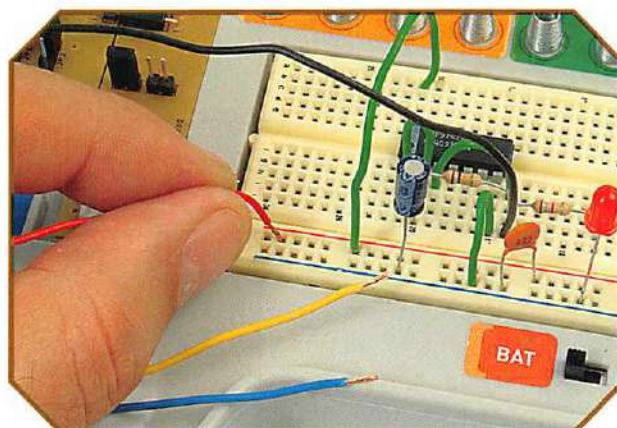
do un ponticello sul connettore JP1 della scheda dei contatori DG03.

### Impostazione a zero

Quando si collega per un istante il filo rosso al positivo, il contatore si cancella e inizia nuovamente a contare da zero.

### Conteggio in binario

Per eseguire questa prova bisogna scollegare C e collegare per un momento A e B. Il contatore avanza e dopo il 9 la cifra delle unità si spegne. Questo avviene perché, pur avendo a disposizione un contatore a quattro bit in grado di contare fino a 15, se le sue uscite sono collegate a un driver che non riconosce le combinazioni dal 10 al 15, e lo zero arriva solo dopo il 15 e con lo zero il riporto che pas-



È sufficiente collegare A per azzerare il contatore.

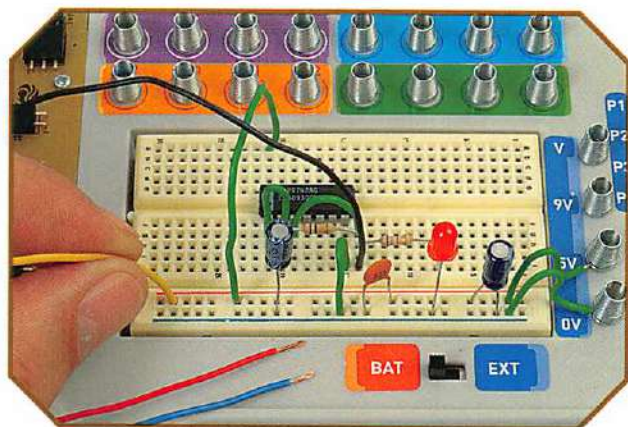
sa all'altro contatore e fa avanzare la cifra delle decine di una unità, è facile dedurre che bisogna scollegare il cavo affinché il contatore possa contare correttamente in modo decimale.

### Ponticello JP2

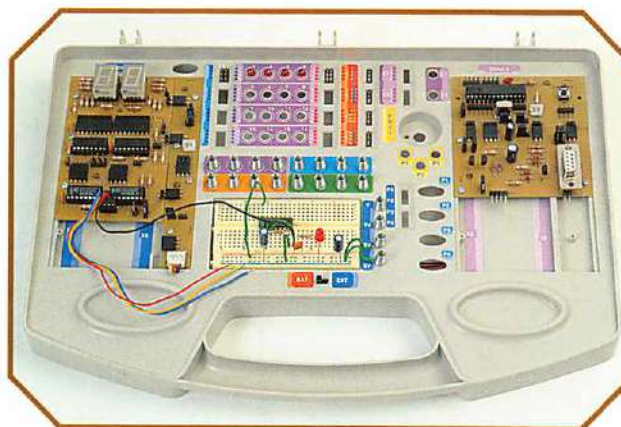
Se togliamo il ponticello e lo mettiamo nell'altra posizione i due contatori avanzeranno in modo simultaneo e la visualizzazione del conteggio sarà 00, 11, 22, 33, 44, ecc.

#### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
R1	Resistenza 100 K (marrone, nero, giallo)
R2	Resistenza 1K8 (marrone, grigio, rosso)
C1, C3	Condensatore 10 $\mu$ F elettrolitico
C2	Condensatore 22 nF
LED A	Diodo LED rosso 5 mm



Collegando B il conteggio è eseguito in binario e la visualizzazione dei numeri sul display è alterata.



Vista del laboratorio con l'esperimento completo realizzato.





## Contatore passo a passo

**È** necessario proteggere i contatori aggiungendo un circuito al loro ingresso di clock per evitare rimbalzi dovuti ai contatti e ottenere così che il contatore avanzi o retroceda di un passo alla volta.

### Il circuito

Con questo circuito si garantisce che a ogni impulso o a ogni collegamento del filo di ingresso rappresentato come A sul circuito, il contatore avanza di una unità. Nel caso non vengano utilizzati circuiti di questo tipo, i cosiddetti rimbalzi dei contatti fanno sì che all'ingresso del contatore arrivino diversi impulsi molto veloci invece di uno solo e dato che il contatore è molto veloce il conteggio avanza

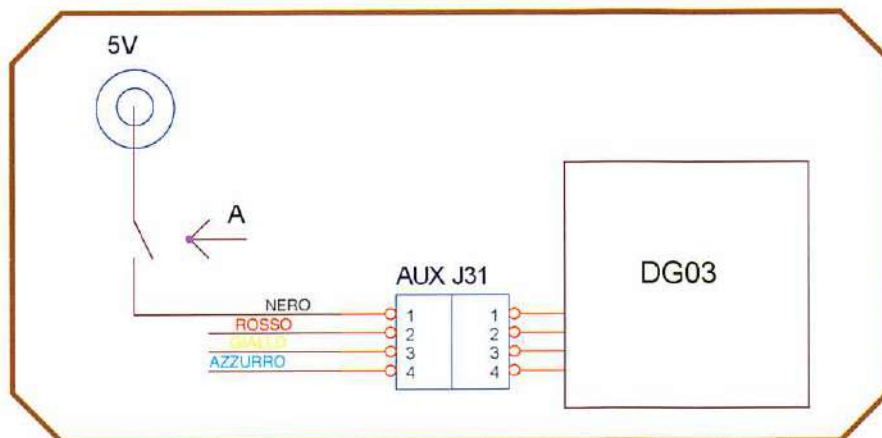
di tante unità quanti sono gli impulsi rilevati. Un pulsante collegato al positivo presenterebbe questo problema, che si potrebbe migliorare utilizzando una resistenza di pull-down, però la soluzione più efficace si ha con l'utilizzo di un circuito monostabile che garantisce l'invio di un solo impulso.

### Il monostabile

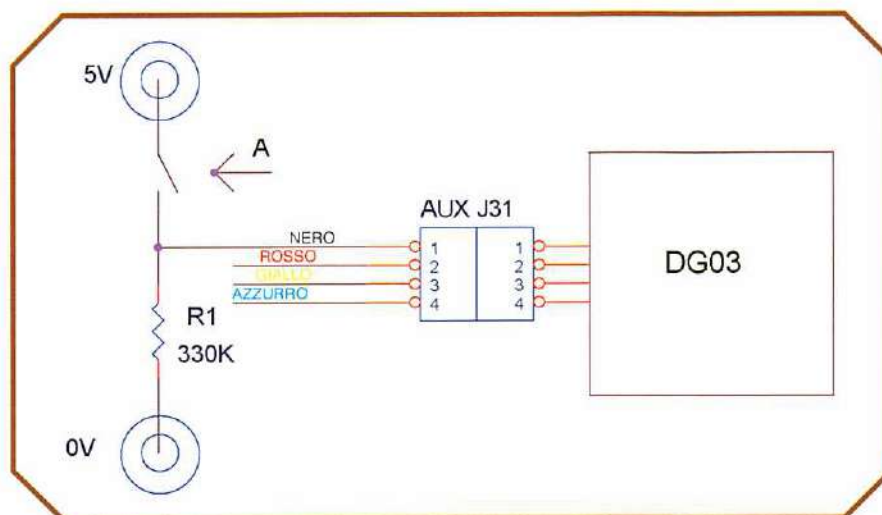
Abbiamo già visto il circuito del monostabile, per questa applicazione però è sufficiente che la durata dell'impulso sia molto breve, per questo motivo il condensatore C3 sarà solamente di 22 nF. Il LED A è utilizzato come indicatore di funzionamento del circuito monostabile, e rimane illuminato per tutta la durata dell'impulso; il collegamento E permette la cancellazione, è sufficiente collegarlo per un breve istante e il contatore si imposta a zero, chiudendo C il contatore conta in senso inverso. Questo collegamento si può anche eseguire utilizzando un ponticello e collegandolo fra i due terminali del connettore JP1 della scheda DG03.

### Montaggio

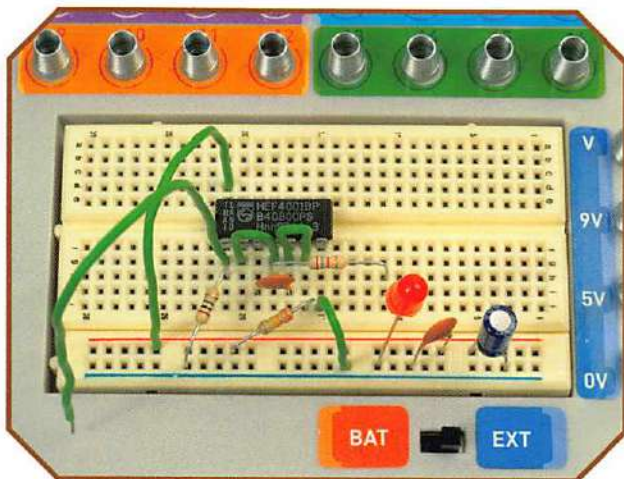
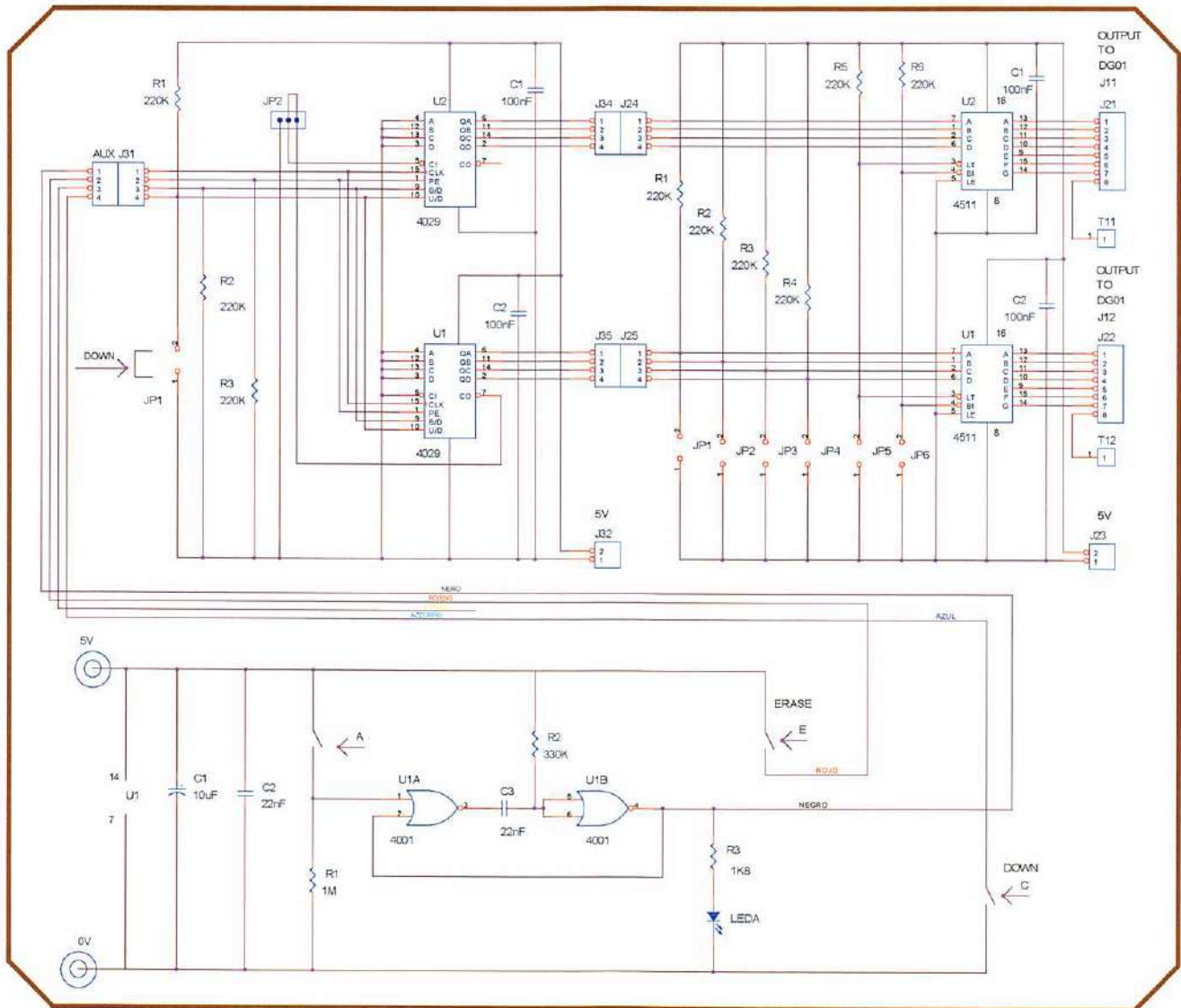
Come ormai d'abitudine prima di realizzare il montaggio bisogna scollegare l'alimentazione, per il momento quindi, il selettore dell'alimentazione deve essere su EXT. Il contatore deve essere completo, ricor-



Collegamento semplice che può produrre rimbalzi.



Collegamento migliorato.



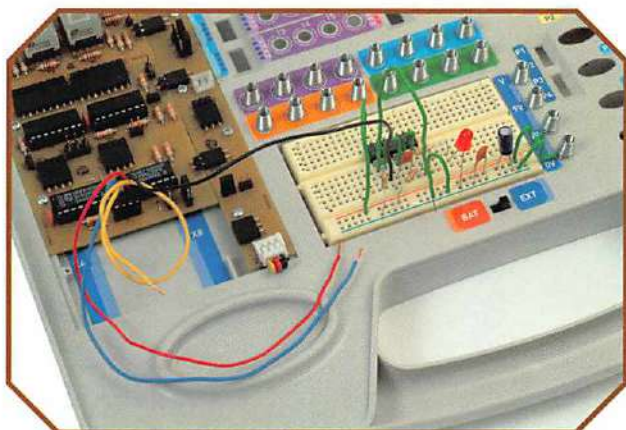
Componenti sulla scheda Bread Board.

#### Schema dell'esperimento.

diamo che è formato dalle schede DG01, DG02 e DG03. Deve essere montato solamente il ponticello JP2 della scheda DG03, non ci deve essere alcun ponticello installato sulla scheda DG02.

I componenti del monostabile, i due condensatori di filtro dell'alimentazione e il LED A con la sua resistenza di polarizzazione verranno montati sulla scheda Bread Board. Bisogna fare attenzione alla polarità del LED A.

Il collegamento alla scheda DG03 si realizza utilizzando un cavetto a quattro fili terminato su un connettore a quattro vie; il connettore di questo cavetto si collega diretta-



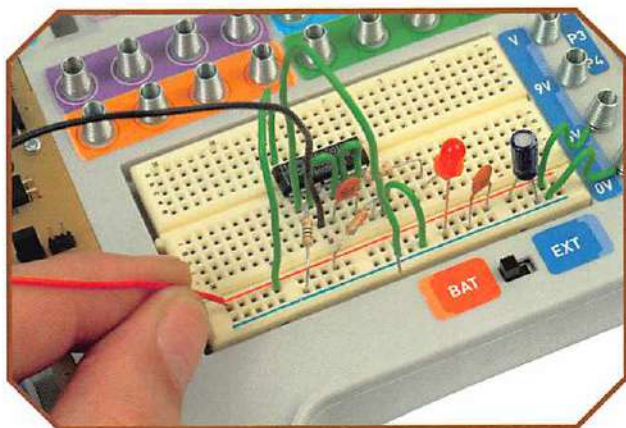
*Cablaggio interno della scheda.*

mente su J31 della scheda DG03 collegando il filo nero al terminale 1, come si può osservare dalle fotografie. Non bisogna dimenticare di collegare il terminale 7 del circuito integrato al negativo dell'alimentazione e il 14 al positivo.

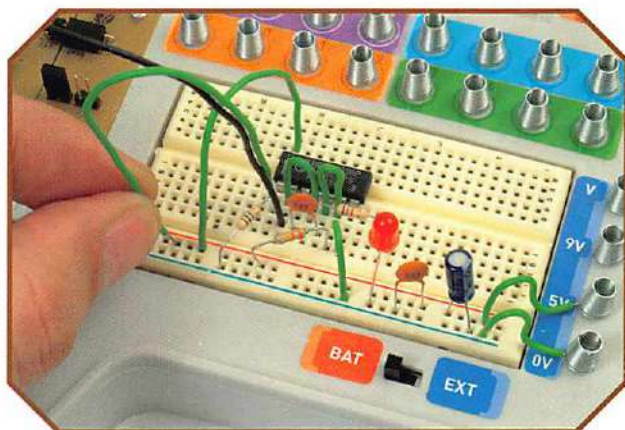
## Alimentazione

Il circuito nel suo insieme può funzionare con qualsiasi tensione compresa tra 4,5 e 12 volt, quindi può funzionare a 5 V o a 9 V, però tutti i circuiti devono essere alla stessa tensione. Ricordiamo che la tensione di alimentazione del contatore si seleziona con i ponticelli delle schede DG04 e DG05.

Per i 5 volt i due ponticelli della scheda DG04 devono essere sulla posizione 1-2, mentre sulla DG05 il ponticello deve essere collo-



*Il collegamento E si utilizza per l'impostazione a zero.*



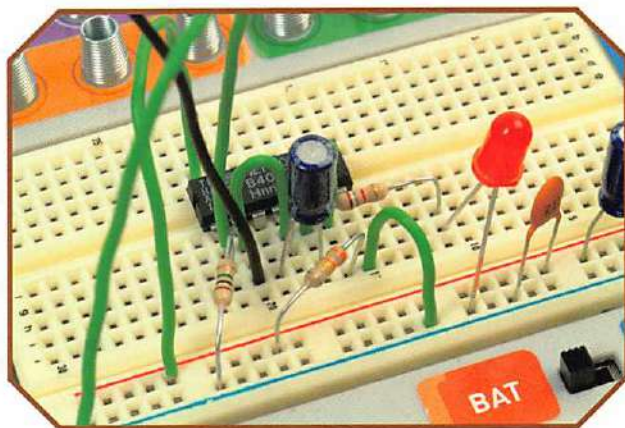
*Ogni volta che si collega A il conteggio avanza di una unità.*

cato su JP1. Per alimentare a 9 volt i ponticelli della scheda DG04 devono essere sulla posizione 3-4, mentre sulla DG05 il ponticello deve essere su JP2. La stessa tensione scelta deve essere utilizzata per il monostabile.

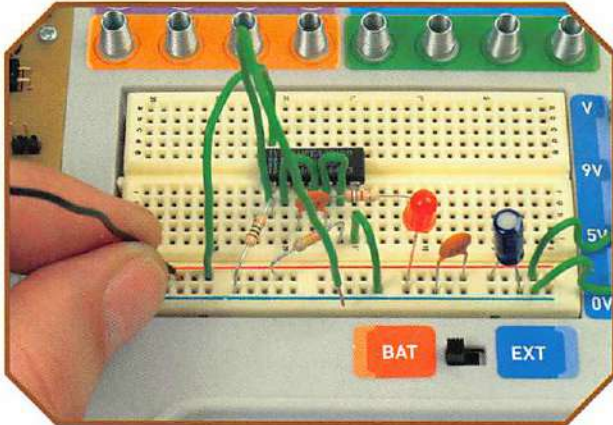
## Prova

La prova inizia collegando l'alimentazione, cioè spostando il commutatore sulla posizione BAT, dato che per ora non disponiamo di un alimentatore esterno. Successivamente collegheremo per un attimo il collegamento E e il display del contatore deve visualizzare la lettura 00.

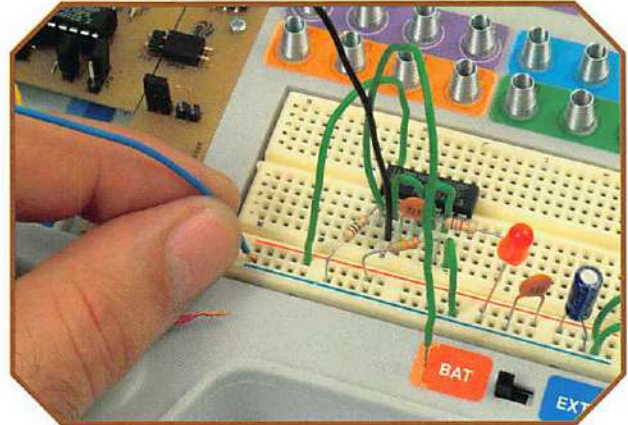
Eseguiamo il collegamento A (quando avremo a disposizione i pulsanti potremo ripetere l'esperimento collegando un pulsante al suo posto) e ogni volta che produrremo un impul-



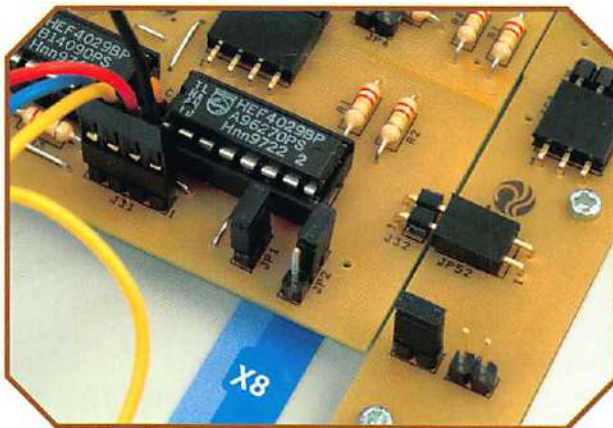
*Con C3 da 10  $\mu$ F si aumenta la durata dell'impulso di uscita del monostabile.*



Se colleghiamo direttamente l'ingresso al positivo possiamo generare dei rimbalzi.



Con il collegamento C chiuso il conteggio è inverso.



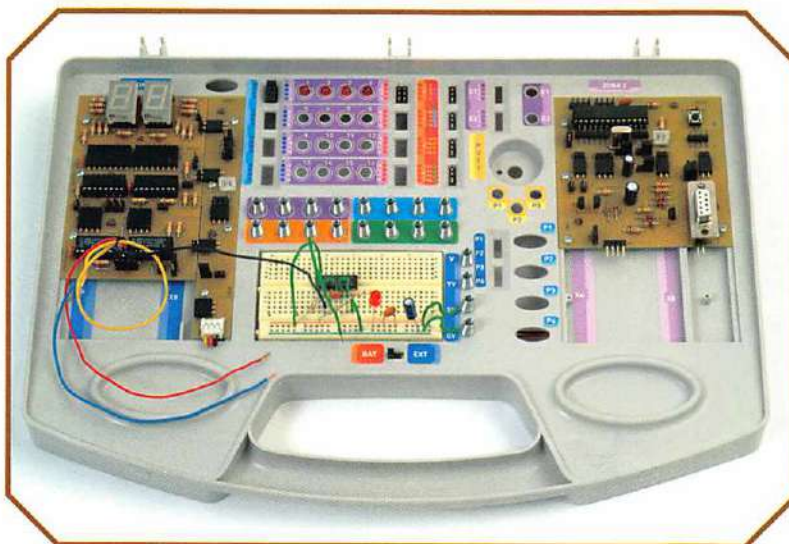
Collegando questo ponticello il contatore conta in senso inverso.

so il conteggio avanzerà di un numero, fino al 99, successivamente passerà a 00.

Se si esegue il collegamento C, il contatore conta in senso inverso e in qualsiasi momento, se si collega E il conteggio torna a zero.

### Variante

È possibile utilizzare un contatore con un tempo di conteggio ancora maggiore, in questo caso il contatore sentirà il nuovo impulso solamente quando terminerà l'impulso precedente; potremo quindi verificare il modo di funzionamento del circuito monostabile applicato all'ingresso del contatore.



Vista del laboratorio con l'esperimento completo.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4001
R1	Resistenza 1 M (marrone, nero, verde)
R2	Resistenza 330 K (arancio, arancio, giallo)
R3	Resistenza 1K8 (marrone, grigio, rosso)
C1	Condensatore 10 $\mu$ F elettrolitico
C2, C3	Condensatore 22 nF
LED A	Diodo LED rosso 5 mm



# Contatore bidirezionale automatico fino a 8

**I**l circuito contatore conta in modo ascendente da 0 fino a 7, poi, automaticamente conta in modo discendente fino a quando il quarto bit passa a 8. A questo punto inizia nuovamente il conteggio ascendente.

## Il circuito

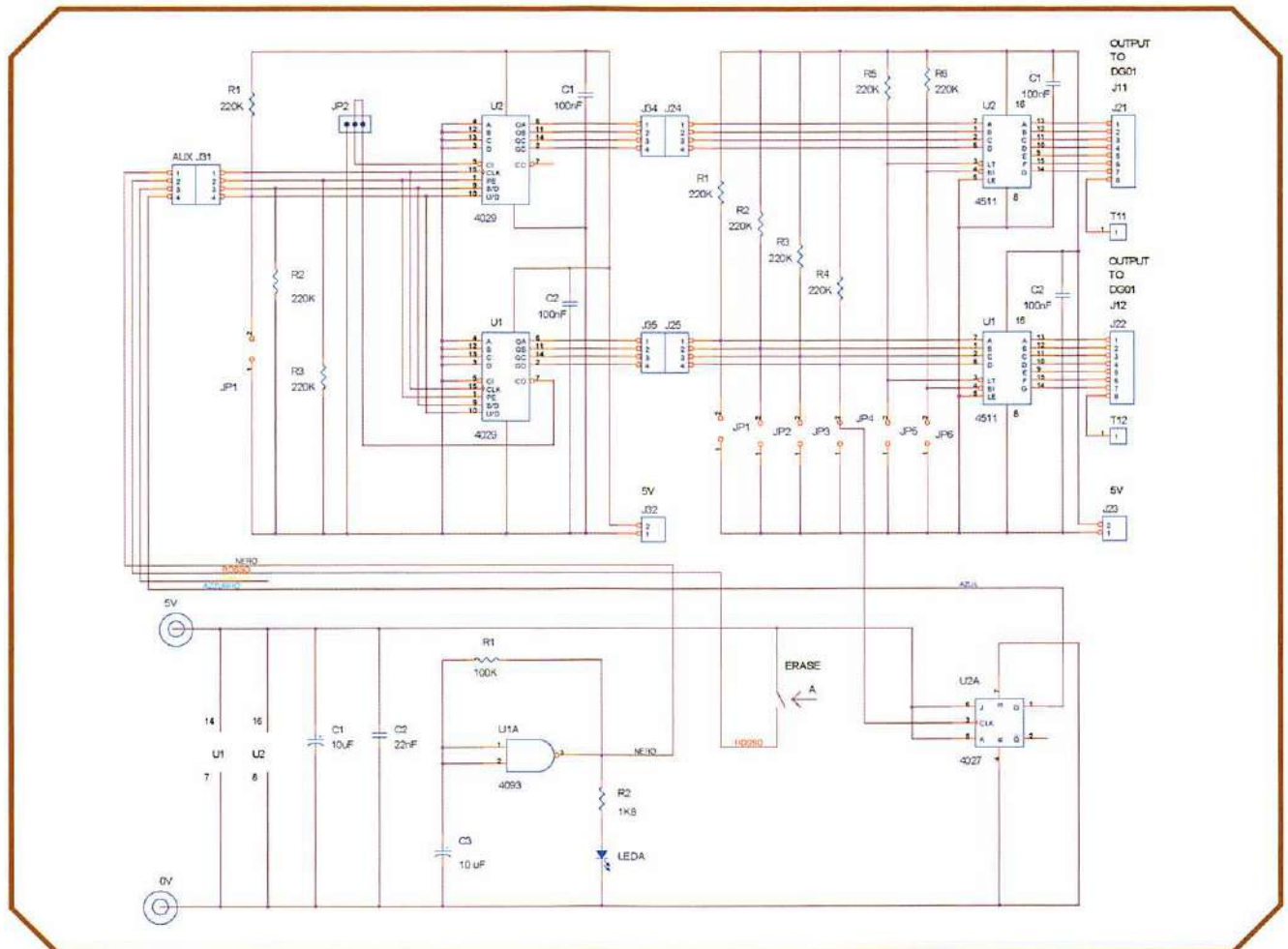
L'obiettivo di questo esperimento è la realizzazione di un circuito semplice con un flip-flop J-K che permette di cambiare in modo automatico il verso del conteggio del contatore a due digit che abbiamo montato nella zona 1 del laboratorio.

La porta U1A sarà il generatore di impulsi applicato all'ingresso del clock del contatore. La frequenza del clock è bassa per poter visualiz-

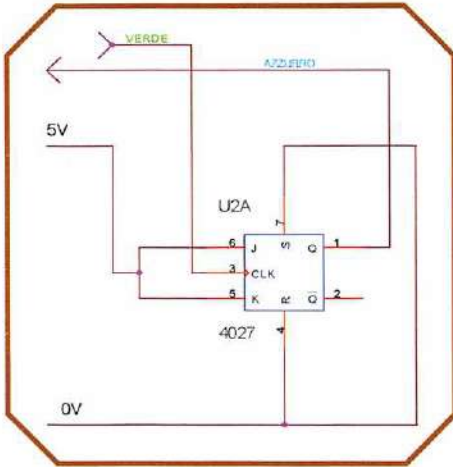
zare sui due display la variazione del conteggio.

Il cambio di direzione automatico si realizza cambiando l'uscita del flip-flop JK e applicandola all'ingresso UP/DOWN del contatore. A questo scopo rileveremo un cambio di livello sul bit D del contatore delle unità U1.

In qualsiasi momento del conteggio è possibile impostare a zero i due display a sette segmenti. Per fare questo effettueremo un Reset dei due contatori azionando il pulsante simulato dal filo A (Erase).



Schema del contatore con inversione su 8.



Dettaglio del collegamento del 4027.

## Il generatore di impulsi - clock -

Il generatore di impulsi è molto semplice, utilizza una porta del circuito integrato 4093, una resistenza R1 da 100 K e un condensatore da 10  $\mu$ F. L'uscita è presa sull'uscita della porta U1A, che corrisponde al terminale 3 del circuito integrato. Lo stato dell'uscita (1 o 0) verrà visualizzato sul diodo LED A. La resistenza R2 limita la corrente che circola sul LED. I condensatori C1 e C2 filtrano l'alimentazione.

## Il flip-flop JK come flip-flop T

Se colleghiamo i due ingressi J e K del flip-flop U2A al positivo dell'alimentazione si ottiene una configurazione di flip-flop molto conosciuta. Questa configurazione di flip-flop JK è nota come configurazione su T.

La particolarità di questo flip-flop consiste nel fatto che ogni volta che arriva un impulso di clock sull'ingresso CLK, l'uscita Q cambia di stato. Quindi se l'uscita è Q=1 e un impulso di clock entra, l'uscita cambierà in Q=0; se entra un altro impulso cambierà nuovamente a 1.

È importante inoltre assicurare che i due ingressi Set e Reset siano disabilitati collegandoli alla massa dell'alimentazione.

## Montaggio

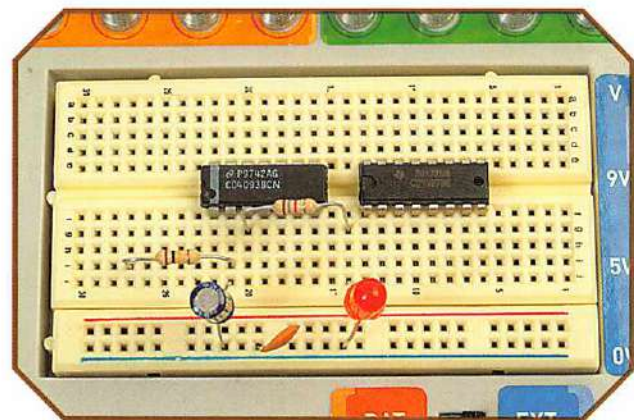
Prima di iniziare il montaggio e durante lo stesso, è necessario essere sicuri che non ci sia alimentazione, per questo vi raccomandiamo di mantenere come ultimo collegamento

quello del positivo dell'alimentazione alla scheda Bread Board.

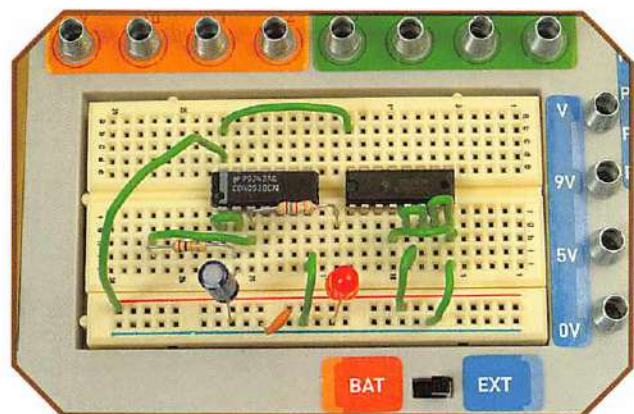
Dato che il contatore è già disponibile è sufficiente montare i componenti del generatore di impulsi sulla scheda Bread Board e realizzare il cablaggio, facendo particolare attenzione alla posizione dei circuiti integrati, alla polarità dei condensatori elettrolitici e a quella del diodo LED.

Il collegamento al contatore si realizza con un cavo a quattro fili terminato su un connettore nero a quattro vie, che si collega direttamente al connettore J31 della scheda DG03, facendo attenzione a collegare il filo nero al terminale 1. Il giallo deve rimanere libero senza nessun collegamento. Non devono essere presenti i ponticelli delle schede DG01 e DG02, invece sulla DG03 bisogna collegare JP2 in posizione normale; potete verificare questa posizione nelle fotografie.

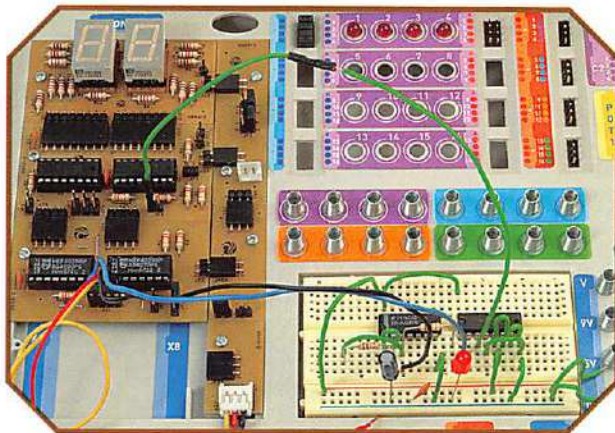
Inoltre bisogna tener presente il collega-



Componenti sulla scheda Bread Board.



Cablaggio interno della scheda.



Collegamento del filo verde su JP4, è il terminale più vicino a J25.

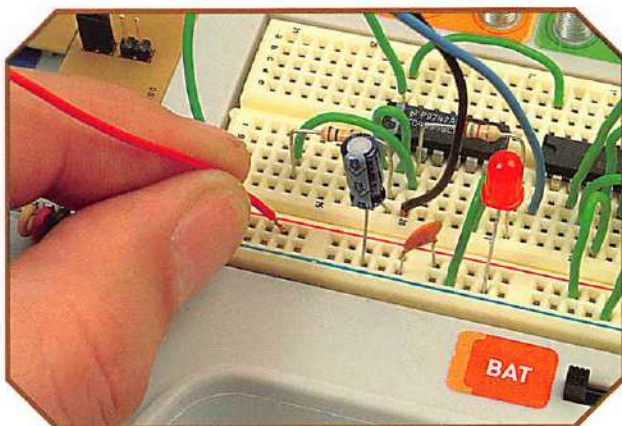
mento del filo verde dal terminale 3 del circuito integrato 4027 fino al terminale JP4 più vicino alla scheda, al connettore J25.

### Alimentazione

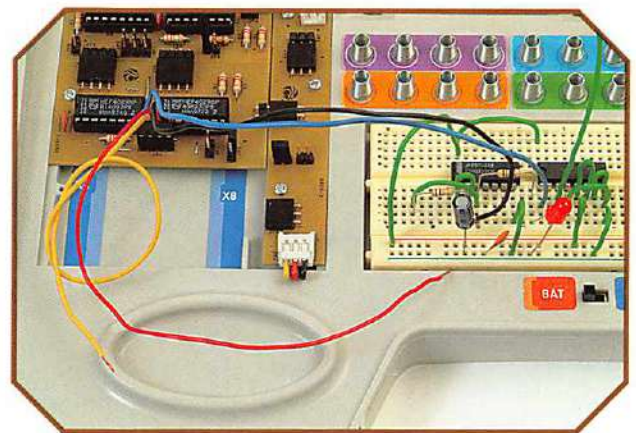
Sia il generatore di impulsi che il contatore possono funzionare a 5 V o a 9 V, però per fare in modo che i livelli di uscita del generatore di impulsi siano compatibili con gli ingressi del contatore, entrambi i circuiti devono essere alimentati alla stessa tensione.

La tensione di alimentazione del contatore si seleziona con i ponticelli delle schede DG04 e DG05.

Per 5 V i due ponticelli della scheda DG04 devono essere in posizione 1-2, mentre sul-



La cancellazione si ottiene collegando il filo rosso al positivo.



Collegamento del cavetto fra J31 di DG03 e la scheda Bread Board.

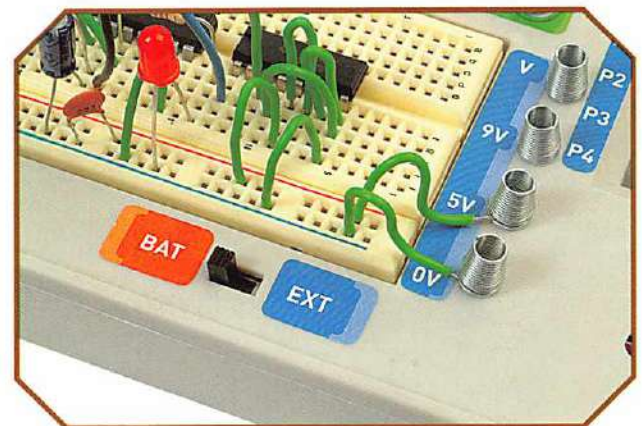
la DG05 il ponticello deve essere montato su JP1.

Per alimentare a 9 volt i ponticelli della scheda DG04 devono essere in posizione 3-4, mentre sulla DG05 il ponticello deve essere su JP2.

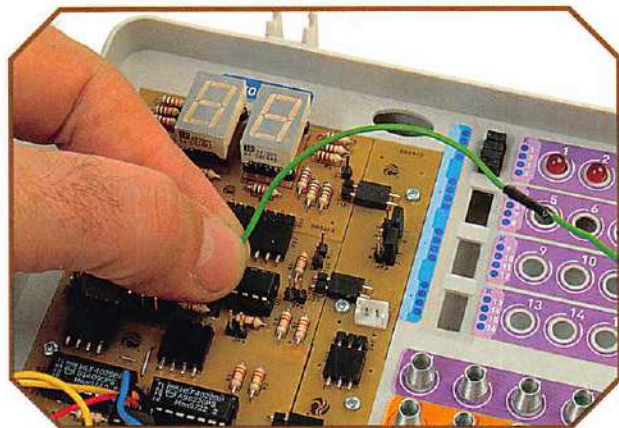
Nel caso dell'oscillatore si utilizzano le molle di alimentazione da 5 V, oppure da 9 V, senza dimenticare di collegare il negativo dell'alimentazione a 0 V.

### Prove

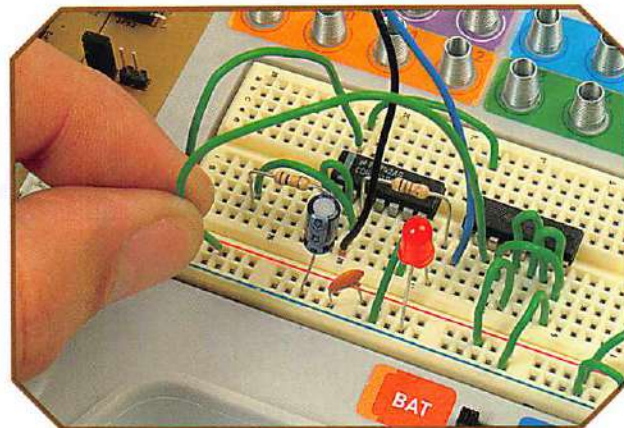
Dopo che il montaggio è stato realizzato in modo corretto è sufficiente collegare l'alimentazione, tutti i circuiti alla stessa tensione e il filo nero del cavetto al terminale 3 del circuito integrato.



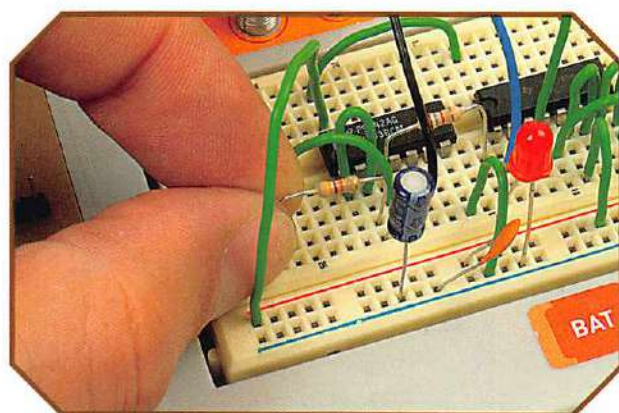
Collegare l'alimentazione a 5 V e commutare su BAT per iniziare.



Staccando il collegamento del filo verde, il contatore non si ferma a 8.



Il filo verde si può utilizzare per invertire il conteggio scollegandolo da JP4 e collegandolo per un attimo al positivo.

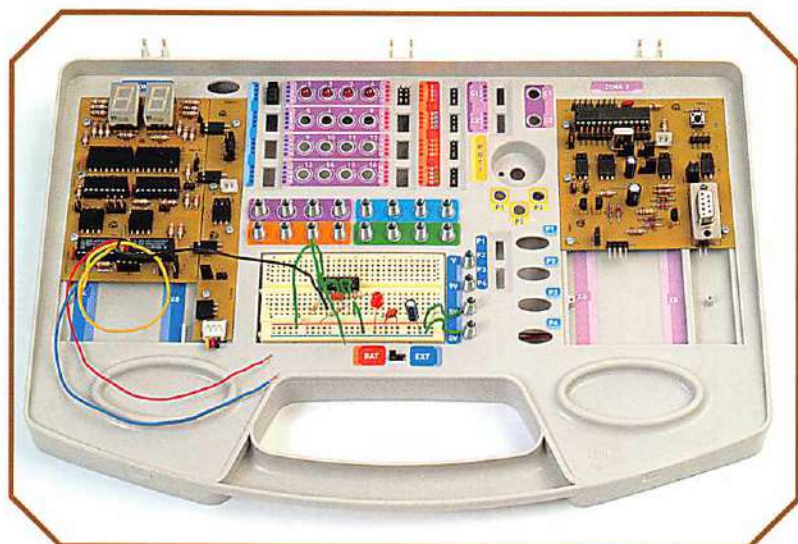


Abbassando il valore di R1 il contatore conta più velocemente.

Collegando l'alimentazione inizierà il conteggio, che sarà ascendente se l'uscita Q di U2A è 1 e discendente se la stessa uscita vale 0.

Collegando l'alimentazione, l'uscita Q può assumere inizialmente un valore qualsiasi. Se ad esempio comincia in verso ascendente, vedremo arrivare i display fino a 07 e quando apparirà 08, inizierà il conteggio in senso inverso, man mano che arrivano gli impulsi di clock, fino a giungere a 98. A questo punto il verso del conteggio cambierà nuovamente e diventerà ascendente.

Se in qualsiasi momento togliamo il filo verde di JP4, il contatore conterà in senso ascendente da 00 fino a 99.



Vista del laboratorio con l'esperimento completo.

### LISTA DEI COMPONENTI

- U1 Circuito integrato 4093
- U2 Circuito integrato 4027
- R1 Resistenza 100 K (marrone, nero, giallo)
- R2 Resistenza 1K8 (marrone, grigio, rosso)
- C1, C3 Condensatore 10  $\mu$ F elettrolitico
- C2 Condensatore 22 nF
- LED A Diodo LED rosso 5 mm





# Esercizio 1: ingressi e uscite, la pratica

**C**ontinuiamo con il progetto di visualizzazione sui LED dello stato delle uscite. Dopo aver realizzato il programma e verificato con la compilazione e la simulazione che non abbia errori, si può scrivere il programma sul microcontroller e passare al montaggio hardware dell'applicazione.

## Configurazione hardware per IC-PROG

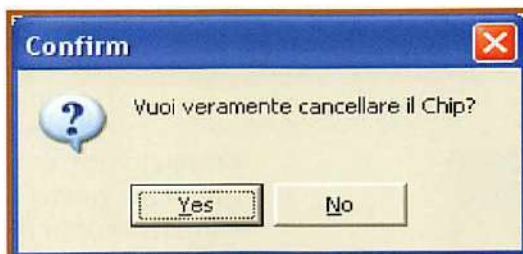
Per scrivere un programma sul PIC quest'ultimo deve essere correttamente montato nel suo zoccolo e i ponticelli (in inglese jumpers) della scheda, devono essere correttamente configurati. Quando si collega il laboratorio al PC per lavorare con IC-PROG, l'alimentazione è fornita dalla porta seriale del PC. Per questo dobbiamo configurare i ponticelli in modo che il laboratorio utilizzi questa alimentazione e i segnali elettrici arrivino correttamente al micro. Ricordate che i ponticelli JP1, JP2 e JP3 devono avere i collegamenti fra i termina-

li 1 e 2 e devono anche essere collegati i ponticelli JP8 e JP9. A questo punto possiamo collegare il cavo seriale tra il PC e il laboratorio.

## Cancelazione

Dopo aver fatto partire il software IC-PROG la prima cosa da fare è leggere il dispositivo per verificare che cosa c'è attualmente sul PIC. Nel caso in cui il contenuto del PIC non sia di nostro interesse, oppure sia scritto con un programma che è già stato salvato nei nostri file, procederemo a cancellare il dispositivo.

Confermeremo la cancellazione leggendo nuovamente il dispositivo e verificando che in tutte le celle della memoria di programma si trovi il valore 3FFF.



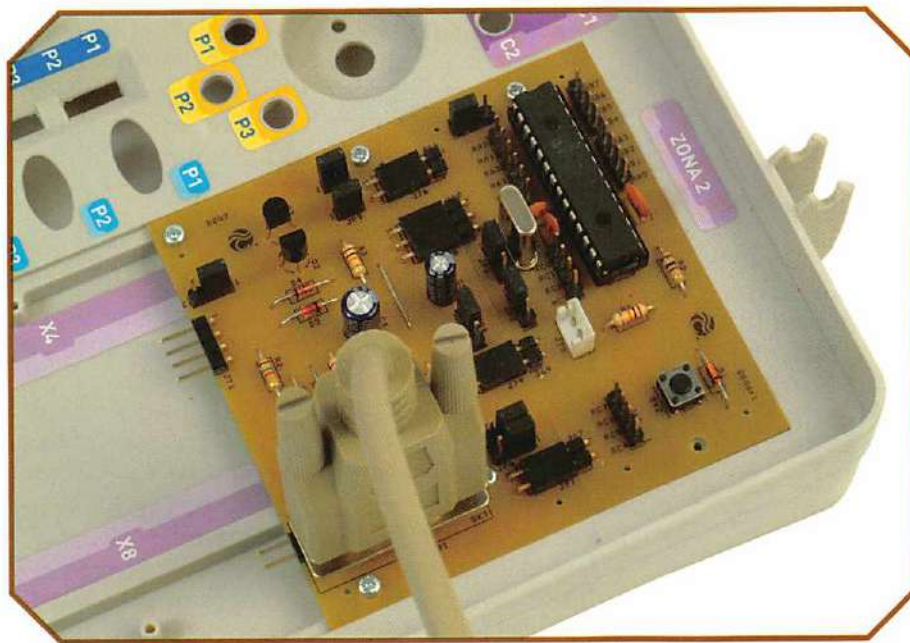
Richiesta di conferma prima della cancellazione.

## Scrittura

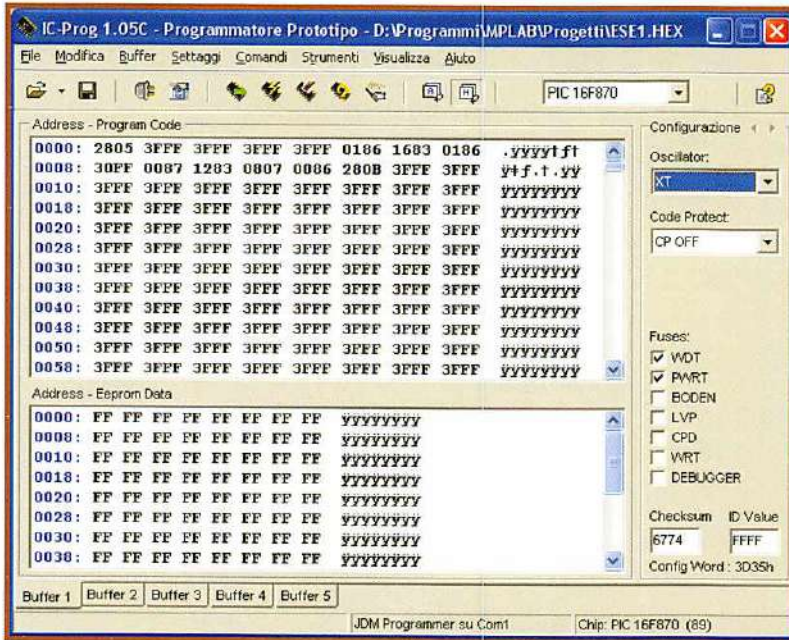
Per caricare il file che vogliamo far girare sul PIC, selezioneremo Apri e caricheremo il file "ese1.hex", però solo se avete in precedenza utilizzato questo nome per salvare il programma che abbiamo realizzato, senza dimenticare che siamo all'esercizio 1. Caricando il file potete vedere come cambia il contenuto delle prime posizioni di memoria.

Dobbiamo configurare i bit della parola di configurazione e l'oscillatore. Selezioneremo un oscillatore tipo XT, e abiliteremo unicamente i bit WDT e PWRT.

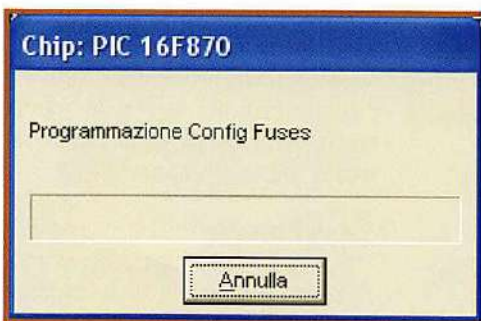
È molto importante mantenere l'opzione di protezione del codice su CP OFF;



Configurazione dei ponticelli per lavorare con IC-PROG.



Aspetto di IC-PROG dopo la configurazione e il caricamento del programma.



Una delle finestre del processo di scrittura.

in questo modo la finestra IC-PROG avrà l'aspetto dell'immagine nella figura.

Possiamo procedere nella scrittura del dispositivo.

Selezionando l'opzione Programma tutto, potremo vedere lo stato delle diverse fasi della scrittura, sino a ottenere alla fine una finestra che ci informa che la verifica è stata effettuata con successo.

Vi consigliamo comunque di leggere il dispositivo per verificare voi stessi che la scrittura è stata effettuata correttamente.

## Configurazione hardware per il funzionamento

A questo punto potete scollegare il cavo fra il laboratorio e il PC. Ora dobbiamo nuovamen-

te configurare l'hardware del laboratorio. L'alimentazione la prenderemo dalle batterie, quindi è necessario cambiare la posizione dei ponticelli. I ponticelli JP1, JP2 e JP3 devono essere collegati fra i terminali 2 e 3, e devono essere tolti i ponticelli JP8 e JP9. In questo modo la configurazione che avrà il laboratorio, sarà quella dell'immagine che raffigura il funzionamento normale.

Abbiamo il PIC scritto correttamente e il laboratorio configurato, quindi possiamo passare alla realizzazione del montaggio che soddisfi le esigenze dell'applicazione.

## Montaggio: ingressi

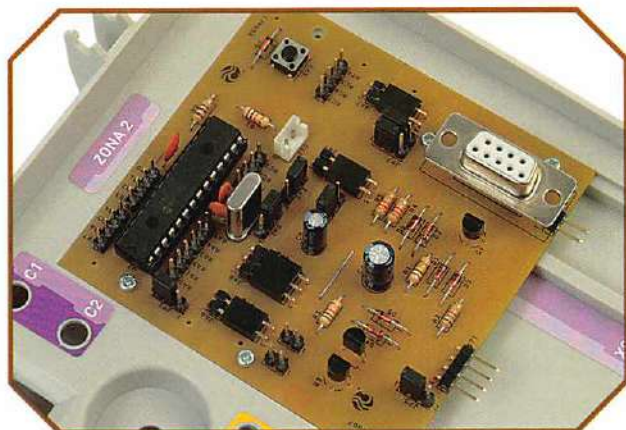
I terminali della porta C del microcontroller sono stati configurati come ingressi, quindi è a essi che dobbiamo collegare i nostri "interruttori". Dato che per provare questo esercizio non è necessario provare gli 8 terminali della porta, inizialmente lavoreremo unicamente con 4 (RC3:RC0). Non abbiamo a disposizione degli interruttori, ma sappiamo che un interruttore discrimina fra due stati: alto o basso. In altre parole dobbiamo applicare agli ingressi del PIC un uno logico (positivo dell'alimentazione) oppure uno zero (0 V).

Colleghiamo un cavetto a quattro fili al connettore J72 per unire l'alimentazione fra le schede e le molle che utilizzeremo come riferimento. È molto importante mantenere il codice dei colori, utilizzando il rosso per il positivo e il nero per 0 V, quindi di questo cavetto utilizzeremo solamente due fili.

Uniremo la molla da 5 V con una fila della scheda Bread Board utilizzando un filo e la molla 0 V con l'altra fila. Da qui otterremo i nostri ingressi per la porta C, come potete vedere nelle fotografie.

Come già sappiamo le molle sono unite internamente ai loro corrispondenti connettori, quindi grazie al cavetto terminato sul connettore possiamo portare l'alimentazione nei punti desiderati delle schede.

È consigliabile utilizzare delle resistenze tra l'alimentazione e gli ingressi, in questo modo proteggeremo il PIC limitando la corrente, e miglioreremo la visualizzazione nella trans-



Configurazione dei ponticelli per il funzionamento normale.

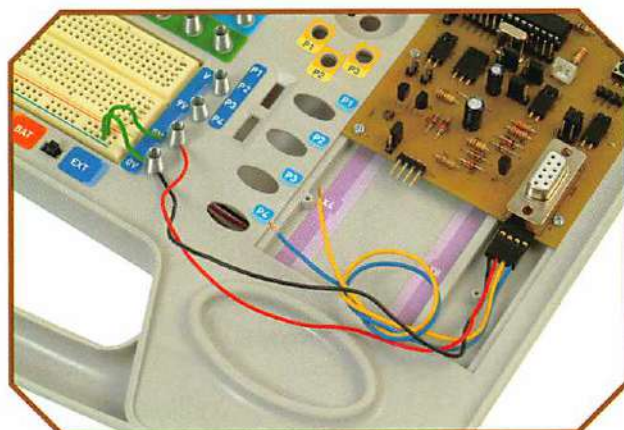
izzazione degli stati. Nelle immagini possiamo vedere il montaggio con le resistenze. Eseguite il montaggio con esse e senza di esse e osservate le differenze.

## Montaggio: uscite

La porta B è stata configurata come porta di uscita, quindi sarà a essa che collegheremo i LED per visualizzare lo stato degli ingressi. Per eseguire questo collegamento collegheremo mediante un cavetto i terminali RB3:RB0 con gli anodi della matrice dei diodi LED, inoltre collegheremo i ponticelli sui catodi.

## Funzionamento

Collegando l'alimentazione al laboratorio potremo vedere come, tramite il PIC, lo stato de-



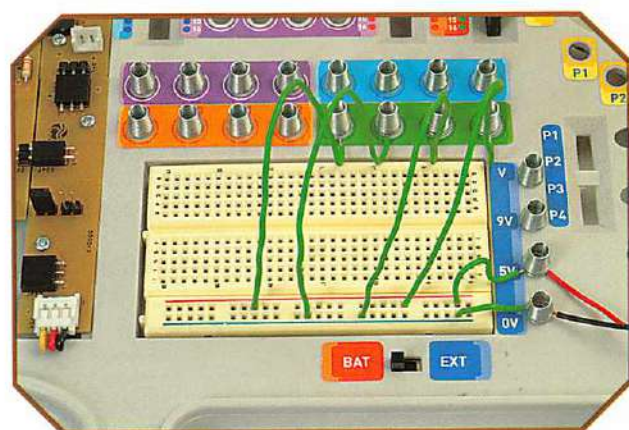
Collegamento dell'alimentazione.

gli ingressi viene visualizzato sui diodi. Se cambiamo gli ingressi tra 0 e 5 V vedremo come si accenderanno e spegneranno i diodi stessi.

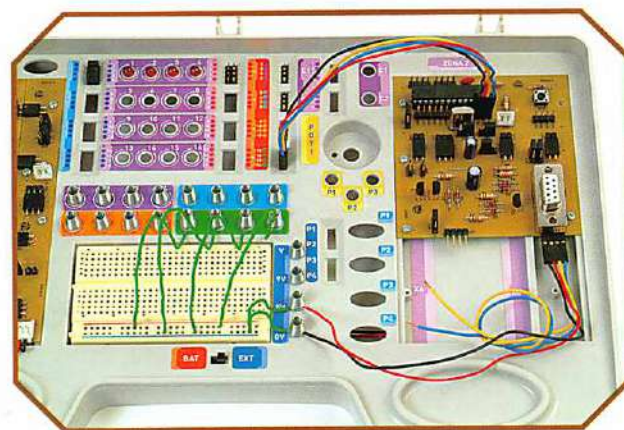
Questa applicazione potrebbe essere utile come parte di un sistema di allarme. Gli interruttori li potremmo sostituire da rilevatori digitali di presenza, di fumo, ecc., mentre i LED funzionerebbero come indicatori per la visualizzazione, oppure potremmo collegare all'uscita un cicalino, con lo scopo di generare un segnale acustico in caso di allarme.

Per terminare la prova del corretto funzionamento dell'applicazione, collegate il cavetto della matrice dei LED ai terminali di uscita RB7:RB4 e il cavetto degli ingressi ai terminali RC7:RC4.

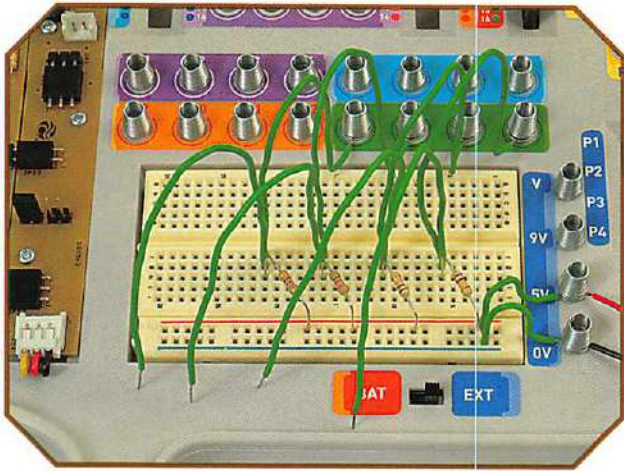
In questo modo verificheremo che anche il



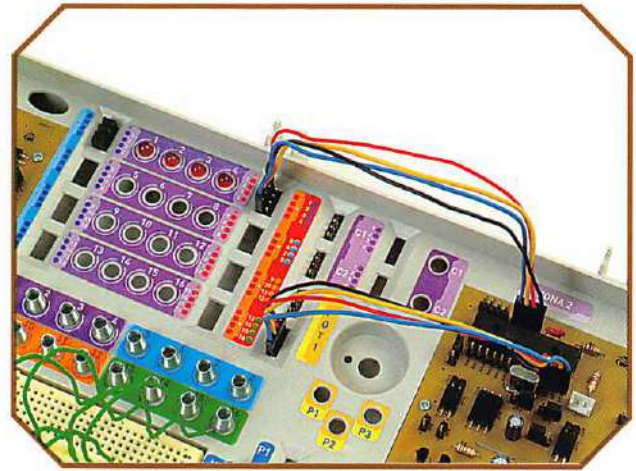
Simulazione degli interruttori.



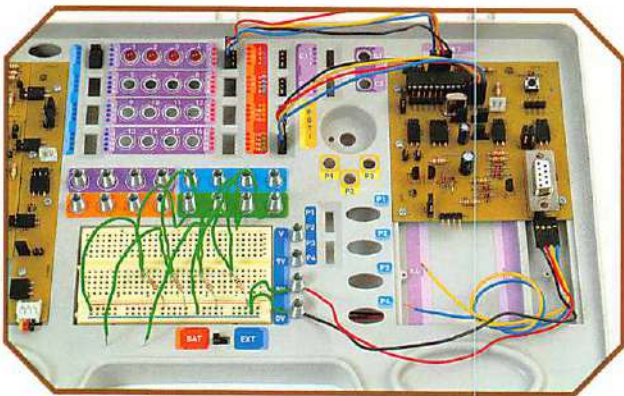
Collegamento degli ingressi.



*Ingressi con resistenze.*



*Collegamenti delle uscite.*

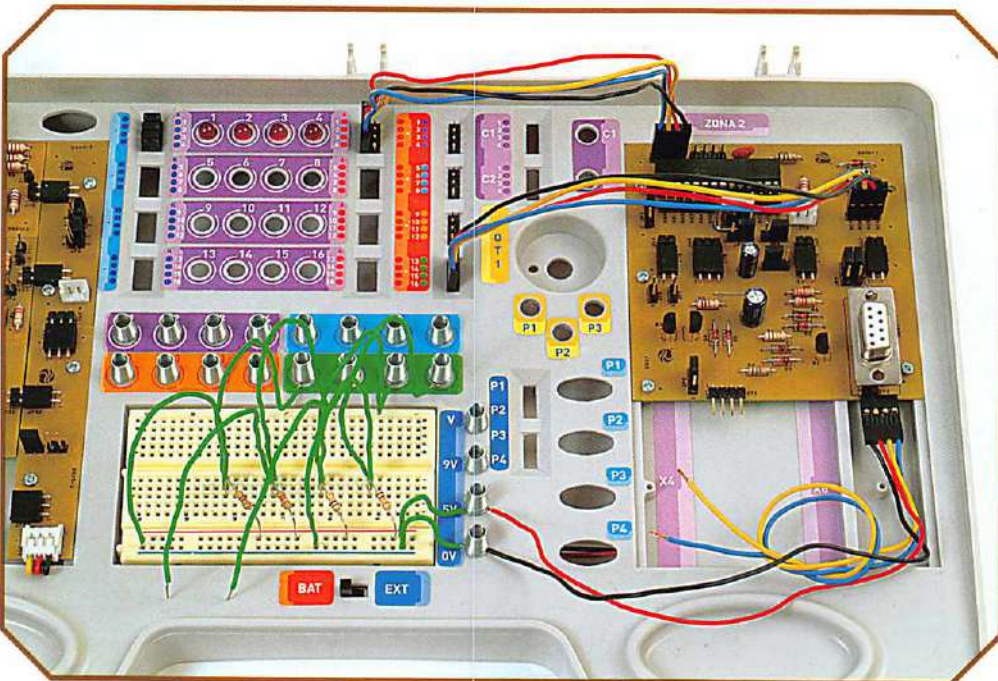


*Vista generale dell'hardware dell'esercizio.*

resto dei terminali di ingresso funzionano correttamente, ottenendo sull'uscita lo stato degli ingressi.

L'applicazione contempla la visualizzazione dello stato di 8 ingressi digitali, con quest'ultima prova possiamo considerare terminato l'esercizio.

Continuando così, poco a poco vedremo salire il grado di complessità degli esercizi che realizzeremo per ottenere il massimo rendimento dal laboratorio e di conseguenza dal PIC16F870.



*Montaggio per la prova degli altri terminali delle porte. Notate che è cambiata la posizione del connettore della porta B, per provare le altre quattro uscite.*



## Esercizio 2: programma combinatoriale, la pratica

**P**er completare l'esercizio 2 dobbiamo fare il montaggio sul laboratorio. Questo montaggio sarà molto simile a quello eseguito per l'esercizio 1, però ci servirà per fare pratica e prendere familiarità con l'hardware. Prima di eseguire il montaggio finale del progetto, dobbiamo caricare il programma sul microcontroller e a questo scopo utilizzeremo IC-PROG. Infine, ci dedicheremo alla soluzione dell'esercizio 3, utilizzando lo stesso procedimento dei due precedenti.

### Hardware per IC-PROG

Supponendo che il PIC sia montato correttamente (inserito nella sua posizione corretta sullo zoccolo) configureremo i ponticelli del laboratorio. JP1, JP2 e JP3 dovranno avere i ponticelli fra i pin 1 e 2 per poter prendere l'alimentazione dal cavo che sarà collegato alla porta seriale del PC.

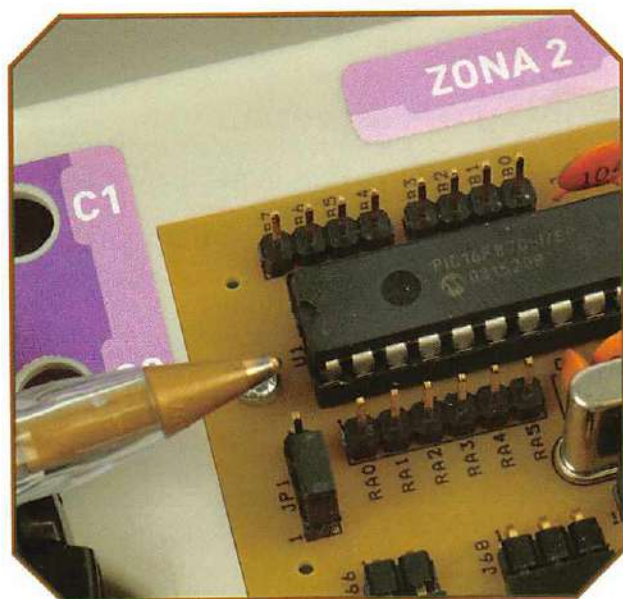
Dovremo inserire i ponticelli JP8 e JP9, che saranno stati tolti se avremo lavorato precedentemente in modo normale o di funzionamento, in altre parole se non avremo utilizzato il laboratorio per la scrittura del microprocessore.

Infine, collegheremo il cavo di comunicazione fra il laboratorio digitale e la porta seriale del PC.

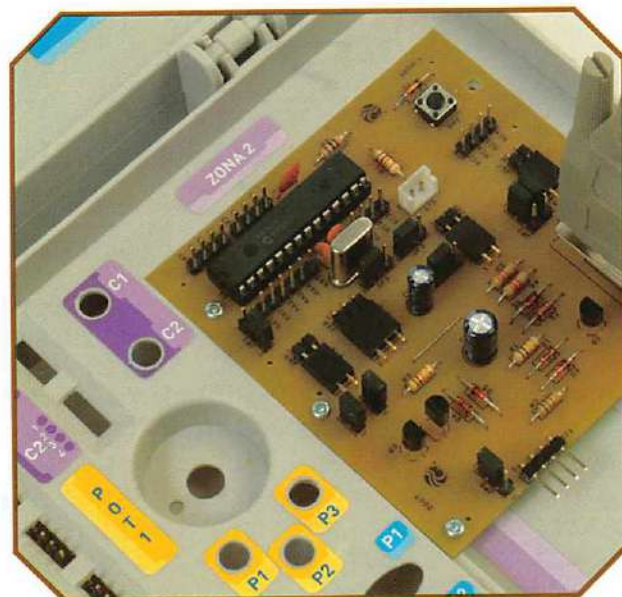
### Cancellazione

Dopo aver configurato l'hardware, siamo pronti per la programmazione del nostro dispositivo mediante IC-PROG. Come abbiamo già detto in altre occasioni è consigliabile cancellare il dispositivo che vogliamo scrivere prima di iniziare l'azione di scrittura. Leggeremo il dispositivo per verificare che non contenga programmi da salvare sul nostro PC, lo cancelleremo direttamente utilizzando il pulsante apposito presente sulla barra degli strumenti.

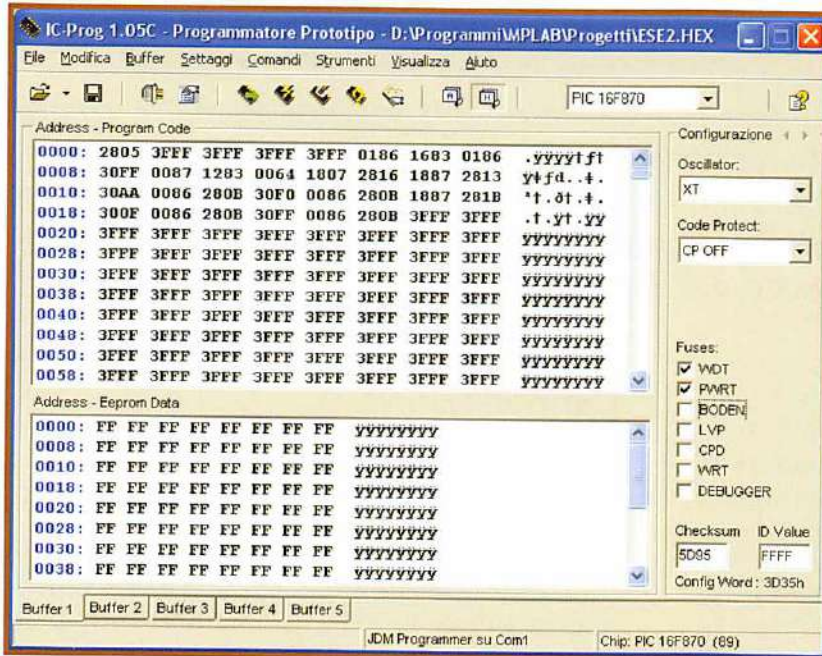
Alla richiesta di conferma di cancellazione, risponderemo affermativamente e verificheremo il risultato dell'operazione, leggendo nuovamente il dispositivo e controllando il contenuto delle posizioni di memoria.



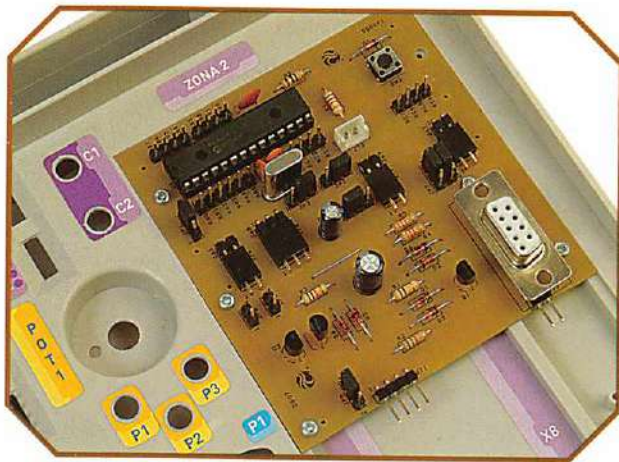
Il PIC deve essere montato correttamente.



Configuriamo i ponticelli per il modo scrittura e colleghiamo il cavo di comunicazione.



Dopo aver caricato il programma IC-PROG dovrà avere questo aspetto.



Il laboratorio con il montaggio finale dell'esercizio.

## Scrittura

Dobbiamo caricare il file che dobbiamo scrivere sul PIC, quindi selezioneremo Apri file, scegliendo il file desiderato. Dopo aver effettuato questo passaggio, la memoria di programma visualizzata sul software, cambierà di aspetto, dato che su essa si troverà il file in formato esadecimale che vogliamo scrivere.

Prima di eseguire il processo di scrittura dobbiamo configurare l'oscillatore e i bit della parola di configurazione. Selezioneremo

un oscillatore di tipo XT, e cliccheremo i bit WDT e PWRT. Ricordate che non stiamo lavorando con un dispositivo dedicato a un programma specifico che non dovrà più essere modificato, quindi bisogna mantenere l'opzione di protezione del codice su CP OFF.

Quando tutto sarà pronto, potremo caricare il programma sul PIC, selezionando il pulsante Programma tutto o premendo il tasto F5. Dopo aver terminato il processo di scrittura, otterremo il messaggio che la verifica è stata effettuata in modo corretto. È consigliabile in ogni caso leggere nuovamente il dispositivo e assicurarci che il programma caricato, corrisponda a quello che volevamo scrivere.

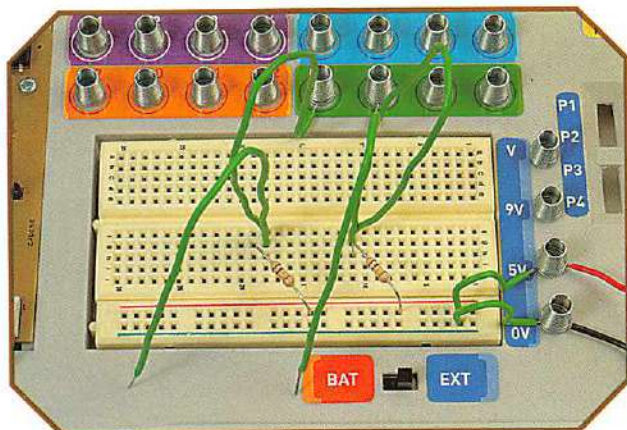
## Configurazione hardware per il funzionamento

Scollegheremo il cavo di comunicazione e configureremo nuovamente i ponticelli del laboratorio per il modo di funzionamento normale. Cambieremo i ponticelli JP1, JP2 e JP3 in modo che ricevano l'alimentazione dalle batterie (pin 2 e 3 con ponticello) e toglieremo i ponticelli JP8 e JP9.

## Montaggio: ingressi

Avremo bisogno di due ingressi che simuleranno gli interruttori e saranno collegati ai pin RC0 e RC1 del microcontroller. Come succedeva con l'esercizio 1, dato che non disponiamo di interruttori li simuleremo mediante due fili, che potranno essere collegati al positivo o al negativo dell'alimentazione. Come abbiamo fatto per l'esercizio precedente, utilizzeremo due resistenze, fra il pin di ingresso e il filo che andrà al positivo o al negativo. Collegheremo anche un cavetto di cui utilizzeremo solamente due fili a J72, per unire le alimentazioni.

Nell'immagine della figura possiamo osservare il montaggio degli ingressi e vedere che differisce dal montaggio dell'esercizio precedente solamente nel numero degli ingressi.



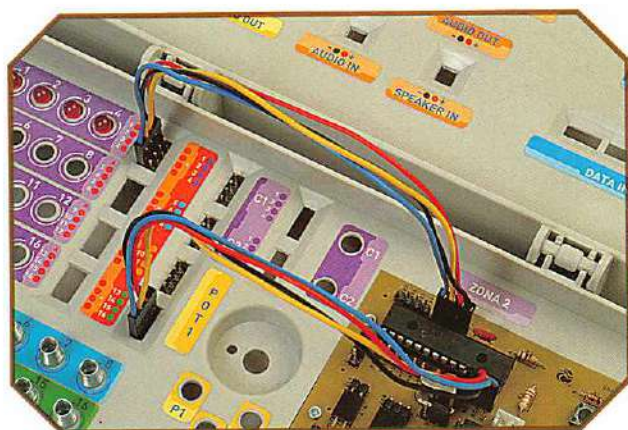
Collegamento degli ingressi sulla scheda Bread Board.

### Montaggio: uscite

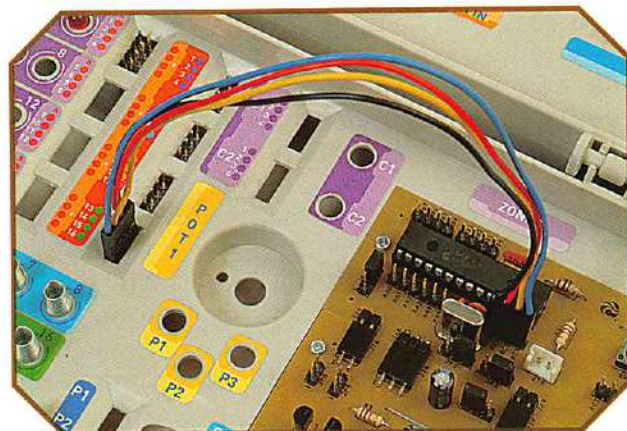
La porta di uscita è la porta B, e anche se la tabella della verità che utilizziamo per assegnare l'uscita contempla l'impiego di 8 bit, noi ne utilizzeremo solamente 4, dato che gli altri 4 saranno una ripetizione dei primi. Collegheremo mediante un cavetto i pin RB3:RB0 con gli anodi della matrice dei diodi e collegheremo i ponticelli sui catodi, come possiamo osservare nella figura.

### Funzionamento

Eseguito il montaggio completo (vedere immagine in basso) collegheremo l'alimentazione e osserveremo il funzionamento. Vedremo come cambiando lo stato degli ingressi, si otterranno combinazioni diverse sulle uscite. Supponiamo un sistema di illuminazione in cui disponiamo di due interruttori e di diverse



Con un secondo cavetto collegheremo le uscite.



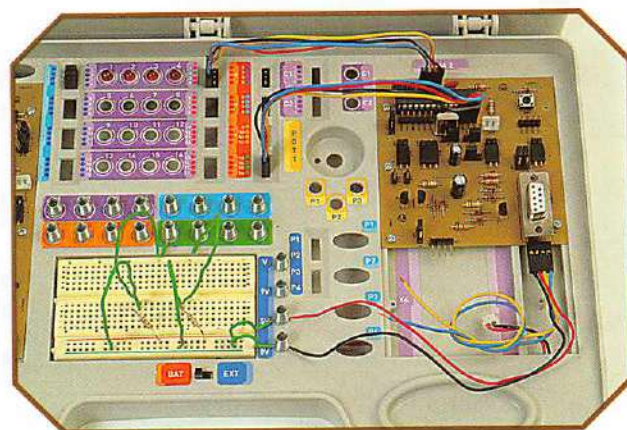
Collegamento degli ingressi sulla scheda DG06.

zone da illuminare. Se il controllo lo realizziamo con un PIC, come previsto in questo esercizio, potremo scegliere combinazioni di zone in funzione dello stato dei due interruttori.

Come ultima cosa verificheremo che sul resto delle uscite otterremo lo stesso risultato. Se colleghiamo il cavetto della matrice dei LED ai pin di uscita RB7:RB4, come mostrato in figura, verificheremo che il risultato è lo stesso del precedente, perché le combinazioni delle uscite sono simmetriche.

### Esercizio 3

Continuiamo con gli esercizi, facendo pratica con gli esempi, come l'esercizio 3 che è contenuto nel CD allegato all'opera, che vi proponiamo di risolvere. Si tratta di una variante dell'esercizio 1, in cui invece di visualizzare sull'uscita il valore dell'ingresso, viene mostrato il risultato della somma di quest'ultimo e di

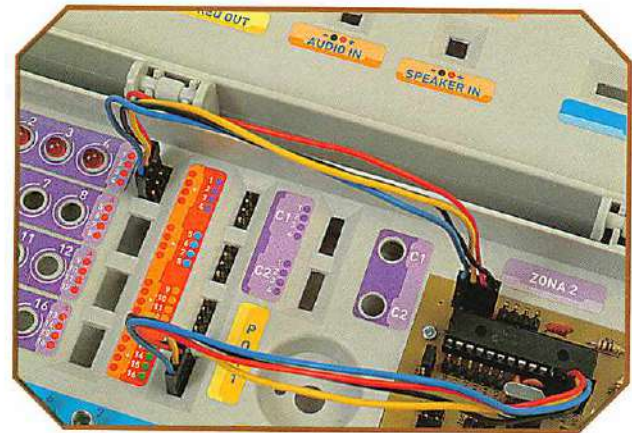


Montaggio completo dell'esercizio.



una costante. Cercate di risolvere voi stessi il programma e ricordate che se utilizzate la soluzione che si trova sul CD, questa non funzionerà, perché è stata pensata per essere scritta su di un PIC in cui è stato precedentemente caricato il programma residente Bootloader. Il montaggio di questo circuito è uguale a quello dell'esercizio 2, tenete presente però che abbiamo un ingresso nuovo (RC2), quindi non risulterà complicata la sua risoluzione hardware.

Per risolvere questo esercizio dobbiamo se-



Verifichiamo che sugli altri pin di uscita il risultato è lo stesso.

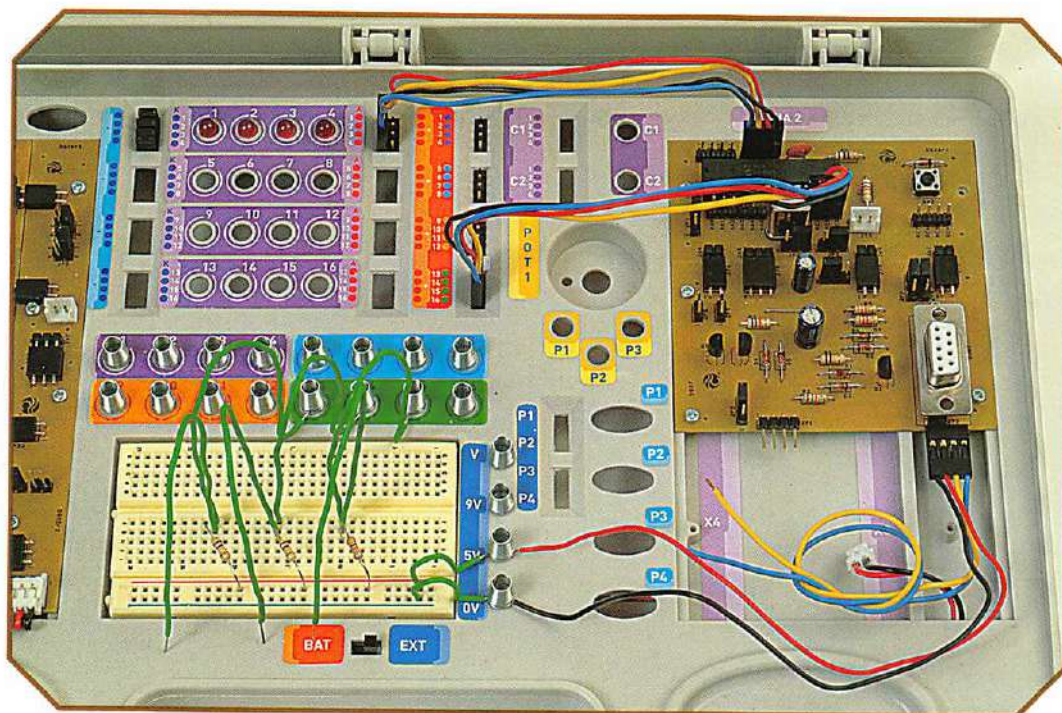
```

Es3 - Blocco note
File Modifica Formato Visualizza ?
:Programma somma
:al valore impostato tramite gli interruttori RC0-RC2 della porta C, si somma una
:costante (in questo esempio il valore 3). Il risultato si visualizza sui led RB0-RB7
:collegati alla porta B

List:  #16F870      ;Processore
       include "16F870.INC" ;Definizione dei registri interni
       org 0x00
Inizio  clrf PORTB      ;Cancella i valori casuali
        bcf STATUS,RP0 ;seleziona il banco 1
        clrf TRISA     ;Porta B si configura come uscita
        movlw b'00001111'
        movwf TRISC    ;3 bit della porta C si configurano come ingressi
        bcf STATUS,RP0 ;seleziona il banco 0
Loop:   clwdt          ;Aggiorna il wdt
        movwf PORTC,W  ;Carica lo stato degli ingressi RC0-RC2
        addlw 3         ;Somma 3
        movwf PORTE    ;Visualizza il risultato su RB0-RB7
        goto LOOP
end     ;Fine del programma
    
```

Codice dell'esercizio 3 che si trova sul CD.

guire gli stessi passaggi visti per i due esercizi precedenti: organigramma, codice, compilazione, simulazione, scrittura sul microcontroller e montaggio. Cambiate il valore della costante nel codice e verificate i risultati ottenuti, realizzate le modifiche pertinenti per la visualizzazione del risultato sul display a 7 segmenti invece che sui diodi LED. In altre parole, giocate con l'esercizio e fate tutta la pratica che potete, sia software che hardware, prendendo confidenza sia con la parte elettronica che con il microcontroller.



Montaggio dell'esercizio 3 è uguale a quello dell'esercizio 2, ma con un ingresso in più, sulla scheda Bread Board.





# Preselezione digitale

**In molti casi è necessario che il conteggio non arrivi al numero più alto che può gestire il contatore. Per questo, è necessario rilevare il numero successivo a quello che vogliamo far contare al contatore. Questa procedura è nota come preselezione digitale.**

## Il circuito

Mediante questo circuito vogliamo realizzare un esperimento con il contatore a due digit montato nella zona 1 del laboratorio. Costruiremo un contatore capace di eseguire un conteggio ascendente fra 00 e 04.

Utilizzeremo a questo scopo, diverse porte NAND per rilevare il numero cinque sul contatore delle unità.

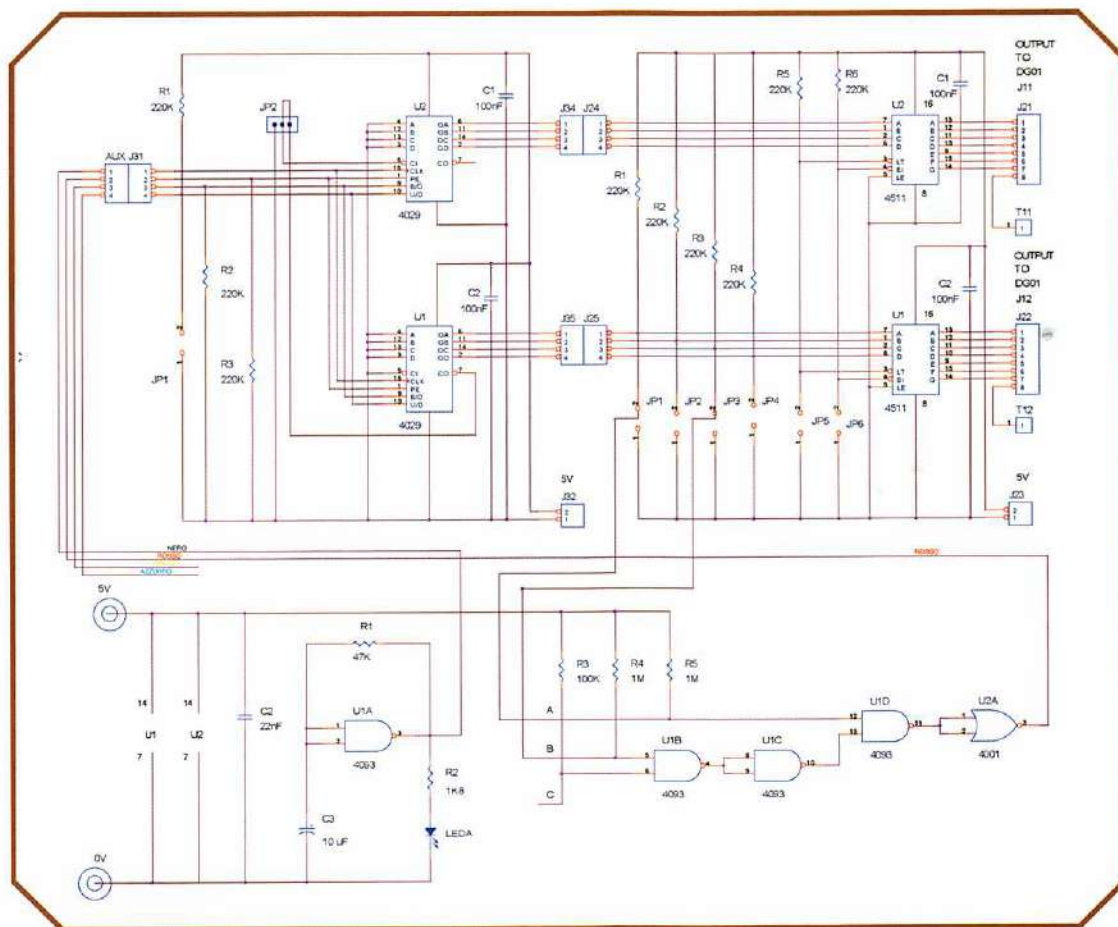
In questo caso, dei due contatori (decine e unità) rileveremo solamente un numero sul contatore delle unità che conterà in modo ascendente. Come massimo quindi, potremo fare un conteggio fra 01 e 08, e come abbia-

mo già detto, in questi due casi dovremo rilevare 02 e 09.

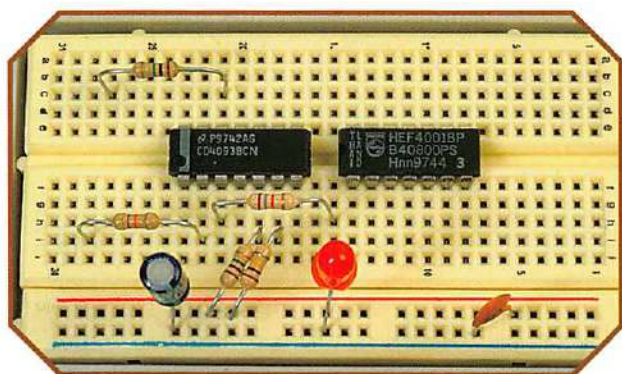
Il riconoscimento del numero successivo al conteggio, verrà fatto mediante le quattro porte NAND. In questo modo è possibile resettare il contatore e farlo iniziare nuovamente da zero.

## Clock

Il generatore di impulsi o generatore di segnale di clock, è una configurazione già nota realizzata con una porta NAND del circuito integrato 4093, una resistenza R1 da 47 K e un condensatore C3 da 10  $\mu$ F. L'uscita è presa sul-



Schema del montaggio completo.



Componenti sulla scheda Bread Board.

L'uscita della porta U1A, che corrisponde al terminale 3 del circuito integrato U1. Lo stato dell'uscita (1 o 0) verrà visualizzato sul diodo LEDA. La resistenza R2 limita la corrente che circola sul LED. Il condensatore C2 filtra l'alimentazione.

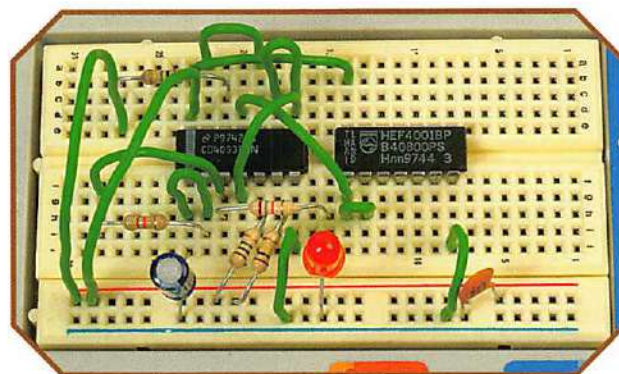
## Preselezione digitale

Non c'è nulla di meglio per capire il funzionamento del circuito che vedere un esempio pratico. Facciamo una preselezione digitale del numero cinque, in modo che il conteggio del contatore sia da 00 a 04.

Per il riconoscimento del numero cinque, rileviamo i due bit A e C che diventano '1' quando appare il numero cinque,  $5 = 0101$ .

Il circuito che abbiamo realizzato è una porta AND a tre ingressi a partire da tre porte AND a due ingressi e una porta NOR che funziona come porta invertente.

Nel caso per il quale abbiamo progettato il



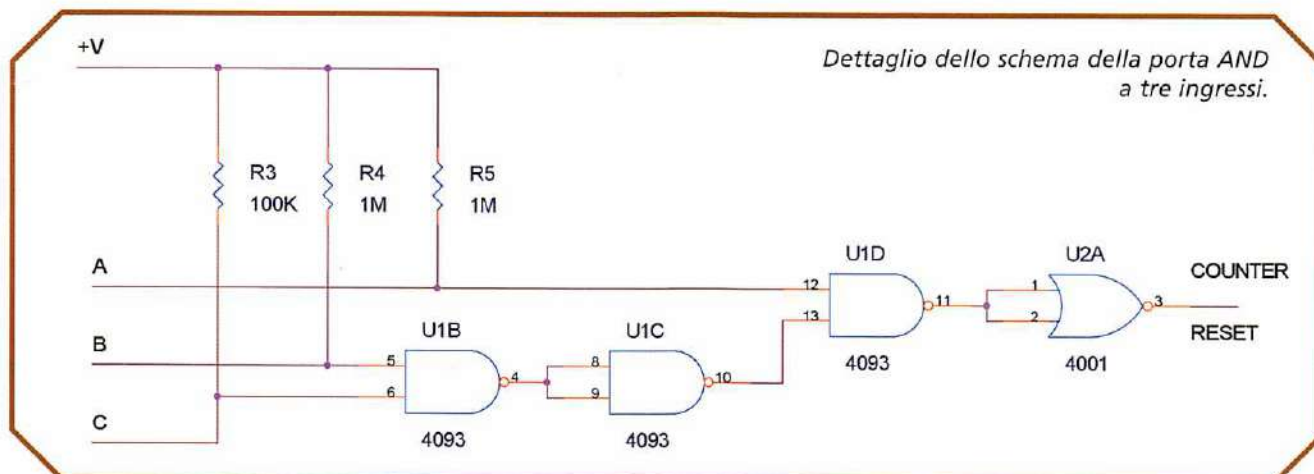
Cablaggio interno della scheda.

circuito, quando appare il numero cinque sul contatore, due dei tre ingressi della porta AND a tre ingressi – quelli segnati come A e B – avranno un livello alto '1' e il terzo lo avrà perché dispone di una resistenza di pull-up, indicata come R3.

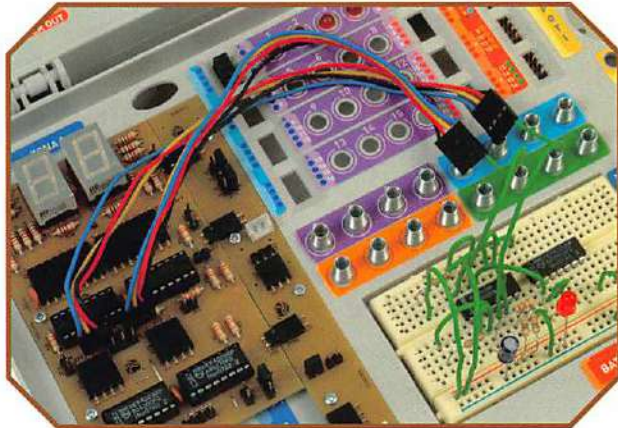
## Il circuito

Se osserviamo lo schema del circuito vedremo sulla parte sinistra l'oscillatore che genera la frequenza di clock. Questo oscillatore è realizzato con una porta NAND del tipo Trigger Schmitt (circuito integrato 4093), perché con una porta NAND convenzionale non oscilla.

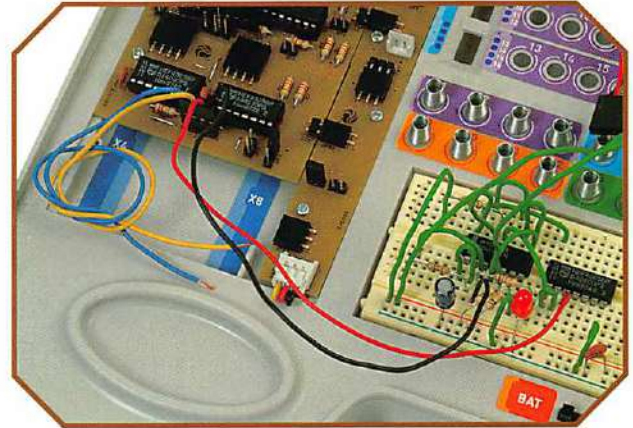
La porta AND a tre ingressi è costruita nel seguente modo: con le porte U1B e U1C si fa una porta AND a due ingressi (NAND seguito da una porta invertente). L'uscita di U1C si applica a uno degli ingressi di U1D. L'altro ingresso di questa porta sarà il terzo della porta AND. L'uscita si deve invertire, quindi utilizza-



Dettaglio dello schema della porta AND a tre ingressi.



Collegamento dei fili su JP1 e JP3. Questo collegamento va effettuato in modo particolare, utilizzando solamente un filo per ognuno dei due cavetti da 4 fili, collegati su JP1 e JP3.



Collegamento del cavetto fra J31 di DG03 e la scheda Bread Board.

remo una porta NOR (U2A) con i due ingressi uniti.

L'uscita si applica all'ingresso PE del contatore per fare in modo che carichi il valore 0000 sulle uscite, in altre parole si esegue un reset del contatore per impostarlo a zero.

## Montaggio

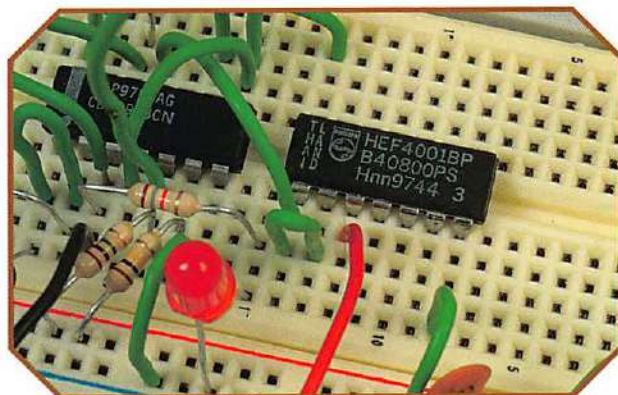
Prima di iniziare il montaggio dobbiamo verificare che l'alimentazione sia scollegata, per questo vi raccomandiamo di lasciare per ultimo il collegamento al positivo dell'alimentazione della scheda Bread Board.

Dato che il contatore è già disponibile, è sufficiente montare i componenti del generatore di impulsi sulla scheda Bread Board ed eseguire il cablaggio, facendo particolare attenzione al montaggio dei circuiti integrati,

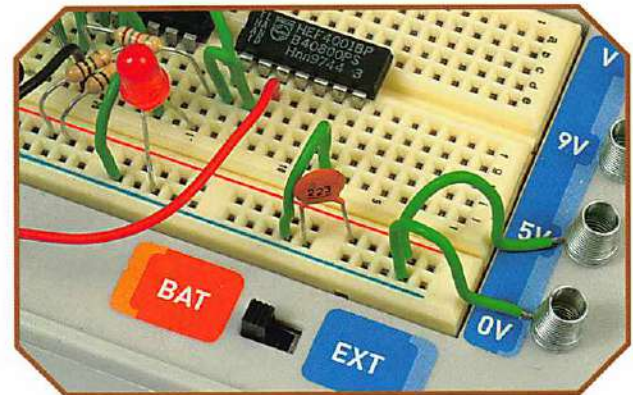
alla polarità dei condensatori elettrolitici, e a quella del diodo LED.

Il collegamento al contatore si realizza con un cavetto a quattro fili, terminato su di un connettore nero a quattro vie, che si collega direttamente al connettore J31 della scheda DG03, facendo attenzione a collegare il filo nero al terminale 1. Il filo rosso si deve collegare all'uscita della porta NOR. Non deve essere collegato nessun ponticello sulle schede DG01 e DG02, invece sulla DG03 bisogna inserire JP2 in posizione normale, potete verificare questa posizione dalle fotografie.

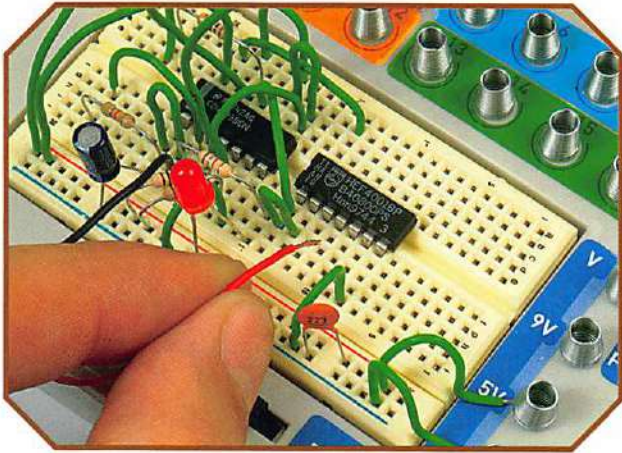
Bisogna anche tener presente i collegamenti fra i terminali 5 e 12 del circuito integrato 4093 sino ai terminali di JP1 e JP3. Questi collegamenti si possono eseguire solamente nel modo che si può osservare nelle fotografie,



Collegamento del filo rosso di reset.



Collegamento dell'alimentazione a 5 V e commutatore su BAT.



Con l'interruzione del filo rosso, il conteggio non si ferma.

anche se si utilizza solamente uno dei quattro fili per ogni cavetto.

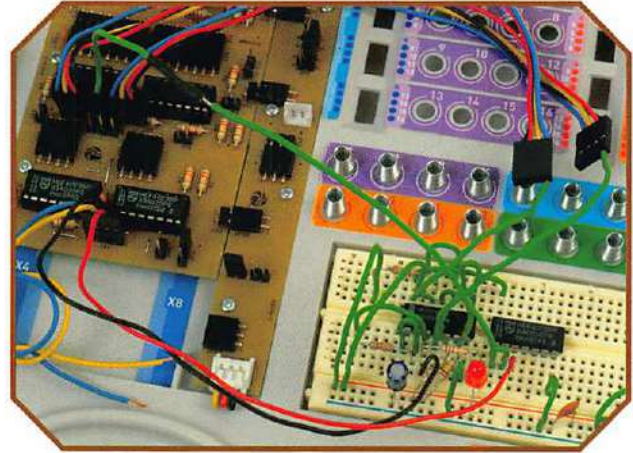
## Alimentazione

Sia il generatore di impulsi che il contatore, possono funzionare a 5 o a 9 V, però per fare in modo che i livelli di uscita dei generatori di impulsi siano compatibili con gli ingressi del contatore, entrambi i circuiti devono essere alimentati alla stessa tensione.

La tensione di alimentazione del contatore si seleziona con i ponticelli delle schede DG04 e DG05.

Per 5 V i due ponticelli della scheda DG04 devono essere in posizione 1-2, mentre sulla DG05 il ponticello deve essere montato su JP1.

Per alimentare a 9 V i ponticelli della scheda DG04 devono essere nella posizione 3-4, men-



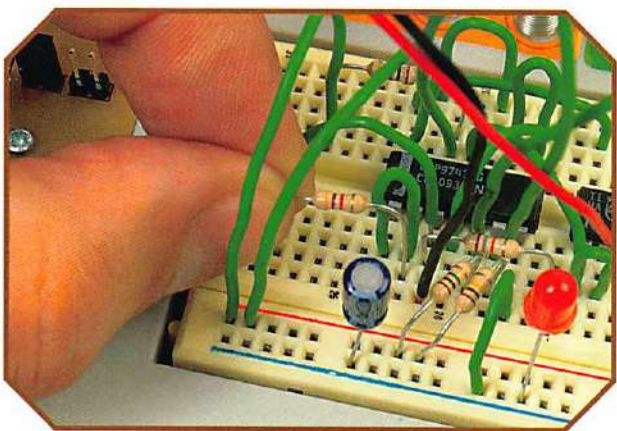
Perchè il conteggio avvenga fra 00 e 06 collegate i 3 ingressi A, B e C della porta AND su JP1, JP2 e JP3.

tre sulla DG05 il ponticello deve essere su JP2.

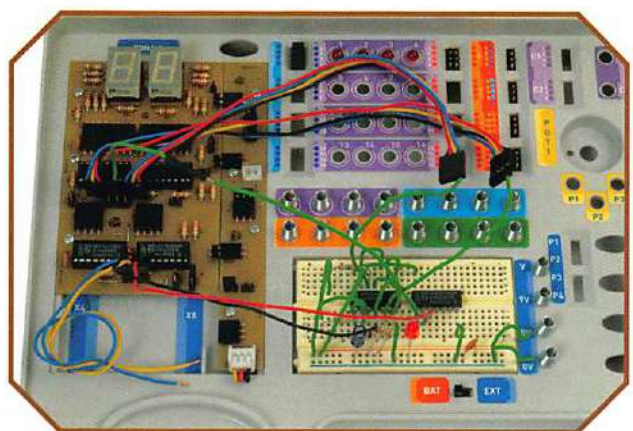
Nel caso dell'oscillatore si utilizza la molla di alimentazione da 5 V o quella da 9 V, senza dimenticare di collegare il negativo dell'alimentazione a 0 V.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
U2	Circuito integrato 4001
R1	Resistenza 47 K (giallo, viola, arancio)
R2	Resistenza da 1K8 (marrone, grigio, rosso)
R3	Resistenza 100 K (marrone, nero, giallo)
R4, R5	Resistenza 1 M (marrone, nero, verde)
C1	Condensatore 22 nF
C2	Condensatore 10 µF elettrolitico
LEDA	Diodo LED rosso 5 mm



Abbassando il valore di R1 il contatore conta più velocemente.



Vista del laboratorio con l'esperimento completato.



## Esercizi 5 e 6: la pratica

**G**iunti a questo punto possiamo eseguire sul laboratorio gli esercizi 5 e 6. I montaggi di entrambi gli esperimenti sono molto simili, quindi vi consigliamo di farli uno di seguito all'altro, dal momento che li vedremo insieme. I due esercizi lavorano con uno dei display a 7 segmenti e simulano degli interruttori agli ingressi del microcontroller. Spiegheremo le parti comuni ai due esercizi senza fare distinzioni, e in seguito, analizzeremo le differenze che ci sono fra di essi.

### Configurazione hardware per IC-PROG

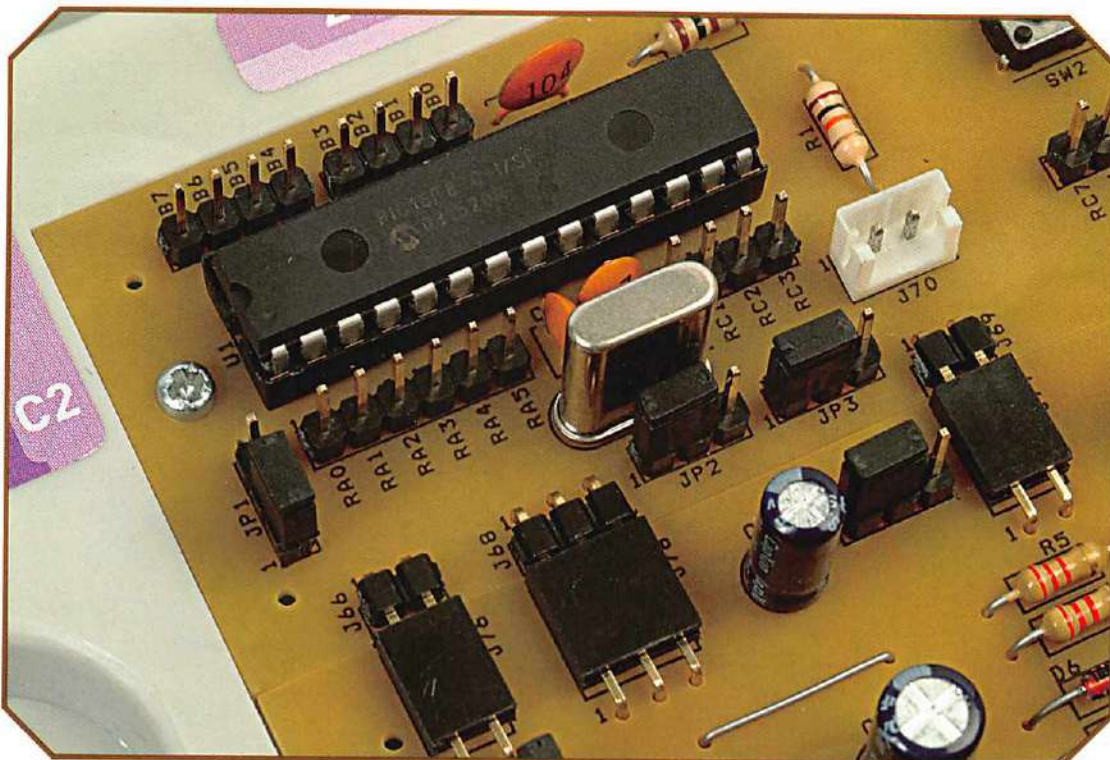
Dobbiamo configurare il laboratorio per eseguire la scrittura del microcontroller utilizzando IC-PROG. Anche se a questo punto voi dovrete già sapere come eseguire questa configurazione, faremo un breve ripasso. Quando vogliamo lavorare con IC-PROG dobbiamo collegare un cavo tra il laboratorio e il PC. Tramite questo collegamento il laboratorio riceverà l'alimentazione dalla porta seriale del PC, quindi dobbiamo predisporre il laboratorio in modo che possa ricevere correttamente un'alimentazione esterna. Ciò si ottiene mediante

i connettori JP1, JP2 e JP3, montando i ponticelli tra i pin 1 e 2.

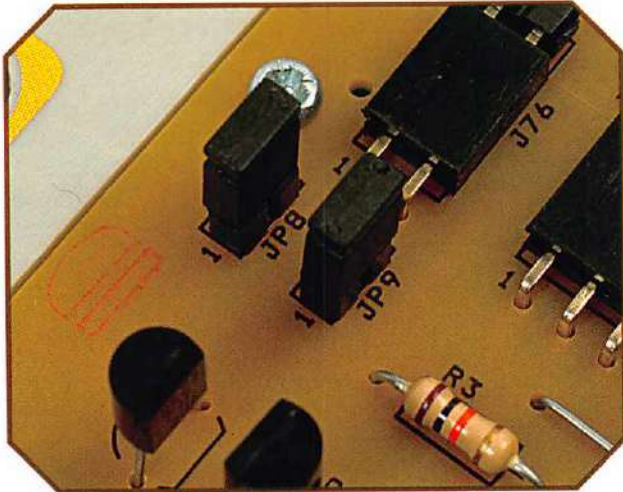
In questo modo si annulla l'alimentazione interna del laboratorio, ovvero quella fornita dalle batterie. Inoltre bisogna anche inserire i ponticelli sui connettori JP8 e JP9 affinché venga realizzato il trasferimento dei dati durante le varie fasi della scrittura. In altre parole, il laboratorio rimane configurato in modo scrittura.

### Cancellazione e scrittura

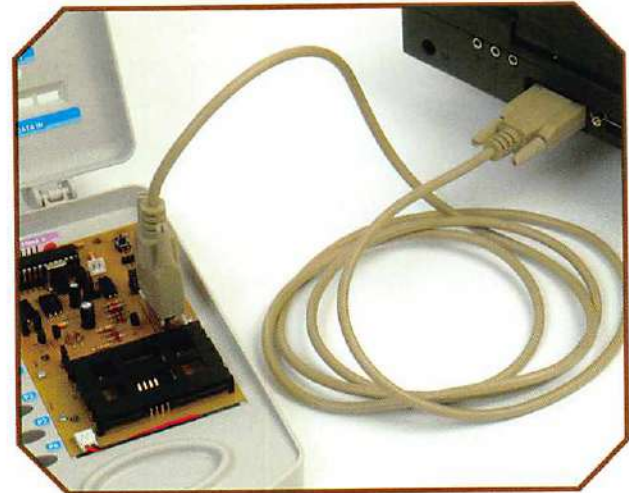
Facciamo partire IC-PROG e carichiamo uno degli esercizi.



Connettori per scegliere l'alimentazione.



Connettori di trasferimento in scrittura.



Cavo di scrittura.

Ricordiamo i passaggi per eseguire questo processo:

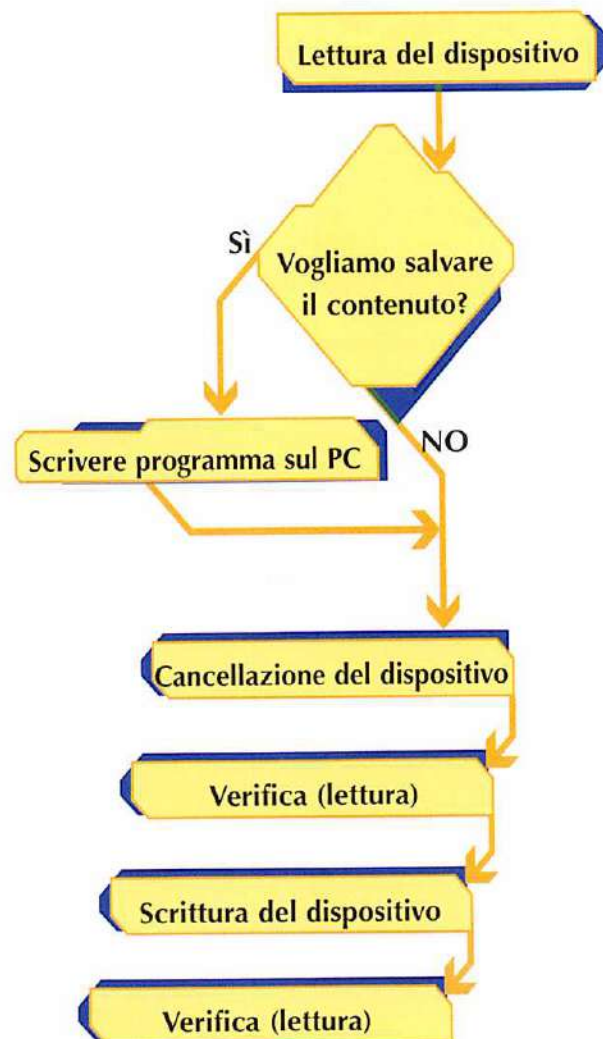
- **Letture del dispositivo:** si deve leggere il dispositivo prima di eseguire qualsiasi azione su di esso nel caso volessimo salvare il programma contenuto nel microcontroller stesso.

- **Cancellazione:** prima di scrivere è sempre consigliabile cancellare il dispositivo per evitare che gli indirizzi di memoria da non modificare nella scrittura contengano qualche dato residuo che potrebbe influenzare il successivo funzionamento.

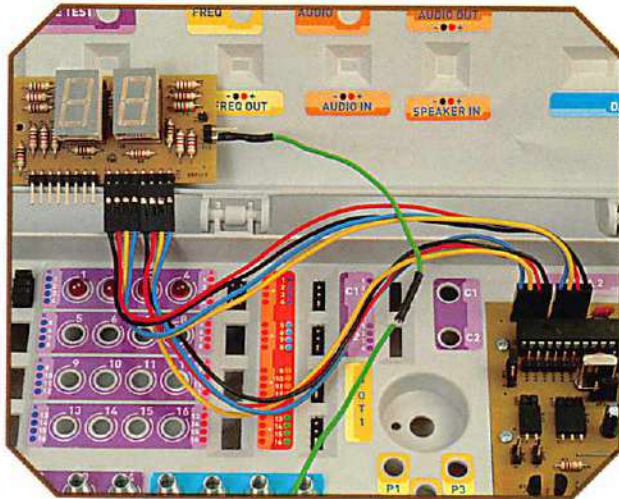
- **Verifica di cancellazione:** anche se il software ci offre la possibilità di verifica, leggeremo nuovamente il dispositivo per verificare che non sia rimasto nulla nelle celle della memoria di programma. Se è così il contenuto dovrà essere 3FFF.

- **Scrittura:** procederemo alla scrittura del dispositivo per il quale avremo in precedenza configurato le opzioni di scrittura su IC-PROG. Scegliremo sempre il tipo di oscillatore XT, la protezione del codice CP OFF, e i bit della parola di configurazione in funzione dei dispositivi che saranno utilizzati nel programma. In questi due esercizi sarà sufficiente abilitare i bit WDT e PWRT, fatto questo potremo selezionare Programma Tutto.

- **Verifica di scrittura:** come ultima verifica del processo leggeremo il dispositivo e controlleremo che contenga il programma desiderato.



Passaggi da realizzare su IC-PROG.



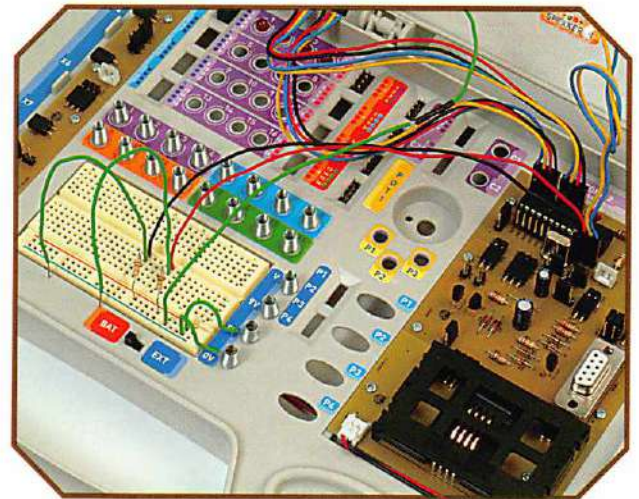
Collegamenti delle uscite verso il display.

### Configurazione hardware per il funzionamento

Passiamo dal modo scrittura a quello normale. Scollegiamo il cavo di comunicazione e configuriamo nuovamente i ponticelli del laboratorio. Prenderemo l'alimentazione dalle batterie, quindi i connettori JP1, JP2 e JP3 avranno i ponticelli tra i pin 2 e 3. Dobbiamo togliere i ponticelli da JP8 e JP9 per liberare i terminali della porta B a cui fanno riferimento.

### Montaggio: uscite

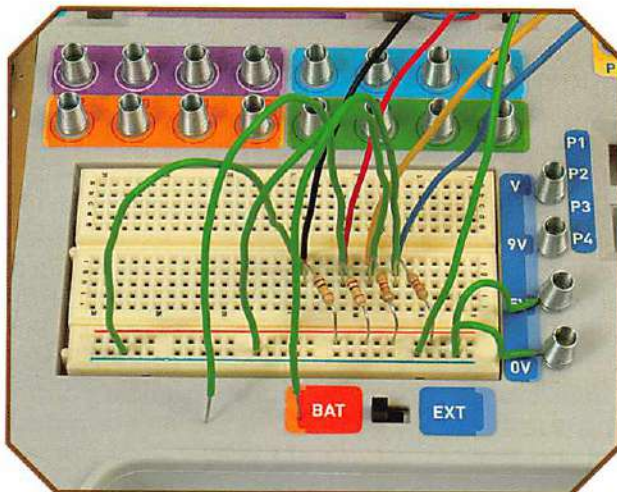
In questi due esercizi il montaggio delle uscite è lo stesso, poiché in entrambi si lavora con il



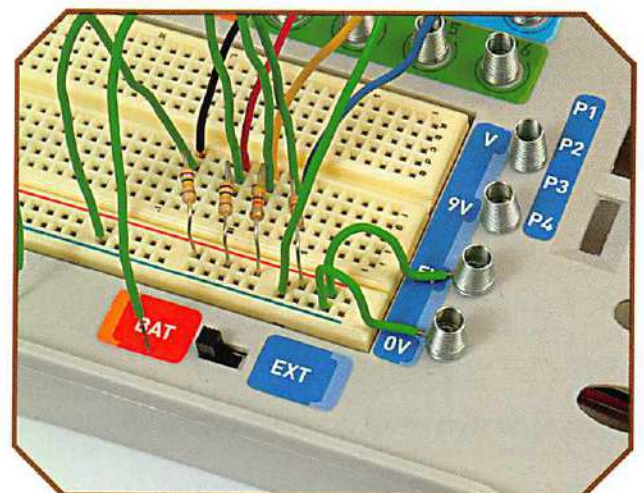
Collegamenti alla scheda Bread Board.

display a 7 segmenti che si trova collegato alla porta B.

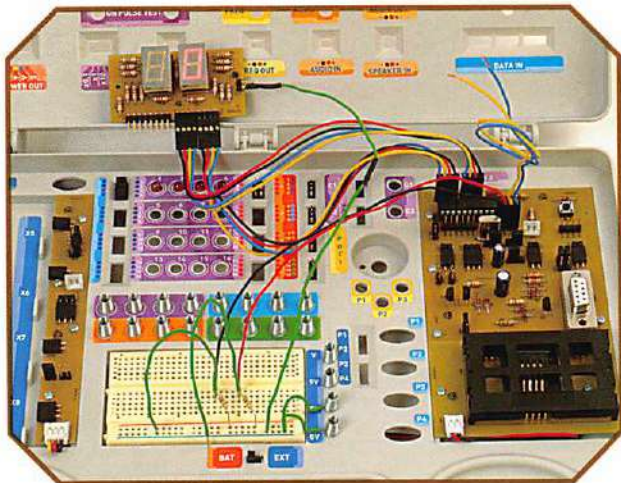
Il circuito stampato DG01 si utilizza isolato dal resto del circuito, quindi dobbiamo smontarlo dal laboratorio togliendo contemporaneamente le schede DG01, DG02 e DG03, per poter successivamente separare la DG01. Utilizzeremo i due cavetti per collegare i terminali della porta B al connettore J12 della scheda DG01 che contiene il display, e un filo per portare il negativo alla scheda DG01, in questo caso sarà di colore verde. Dato che l'uscita del PIC è direttamente in codice a 7 segmenti, non sarà necessario passare attraverso i driver per adattare il segnale al display. Nelle immagini si può vedere come realizzare questi



Collegamenti degli ingressi per l'esercizio 6.



Possiamo collegare le alimentazioni.



Montaggio finale per l'esercizio 5.

collegamenti. È molto importante non incrociare i fili uno con l'altro fra di loro, allo scopo di mantenere la corrispondenza tra i pin dell'uscita del microcontroller e quelli del display.

### Montaggio: ingressi esercizio 5

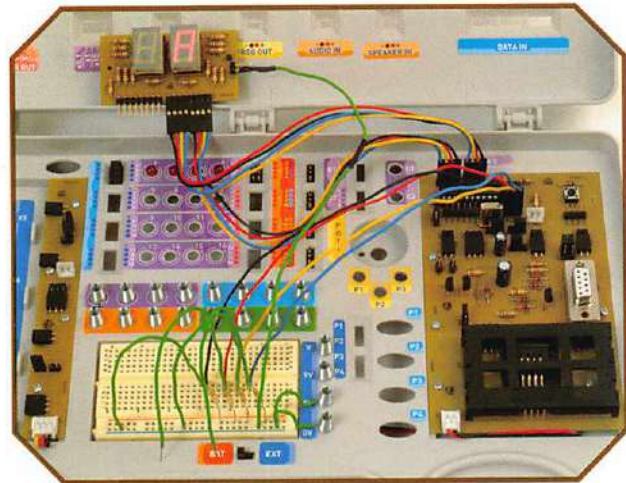
Abbiamo bisogno di due ingressi che simulano gli interruttori e saranno collegati ai pin RC0 e RC1. Negli esercizi precedenti abbiamo imparato che la simulazione degli interruttori è tanto semplice quanto utilizzare un filo da collegare al positivo o al negativo dell'alimentazione. In questo modo contemperemo i due possibili stati dell'ingresso. È consigliabile utilizzare resistenze tra l'alimentazione e i terminali di ingresso. Nell'immagine potete vedere i collegamenti degli ingressi.

### Montaggio: Ingressi esercizio 6

Il montaggio delle uscite di questo esercizio è molto simile al precedente, con la differenza che ora lavoriamo con quattro ingressi, avendo bisogno di quattro digit binari da convertire in codice esadecimale. Dovremo solamente ampliare il nostro montaggio con due nuovi ingressi, con i rispettivi fili e le resistenze, come si può vedere nella figura.

### Funzionamento esercizio 5

Dopo aver eseguito il montaggio completo (vedere l'immagine in alto a destra) collegiamo



Montaggio finale per l'esercizio 6.

mo l'alimentazione e verifichiamo il funzionamento. Se all'inizio gli ingressi si trovano a 0 Vdc (questo dovrebbe essere lo stato iniziale del sistema) il display visualizzerà uno 0. Proviamo ad attivare il punto decimale e in seguito cambiamo di stato gli ingressi. Immaginate un sistema di rilevazione di camera occupata in un hotel. Lo 0 potrebbe significare camera libera e l'1 camera occupata. Il punto decimale potrebbe significare il desiderio dell'occupante di non essere disturbato.

### Funzionamento esercizio 6

Se aggiungiamo i due ingressi per poter eseguire questa applicazione e abbiamo il programma corrispondente che gira sul PIC, possiamo verificare che all'inizio gli ingressi saranno tutti a 0, quindi il display visualizzerà uno 0. Dobbiamo eseguire il conteggio degli ingressi in binario per poter vedere che a ogni ingresso corrisponde un numero in esadecimale. Immaginate un sistema di decodifica qualsiasi in cui si vogliono tradurre dei simboli in un linguaggio più comune. Il sistema di lavoro sarebbe simile, una tabella per assegnare un carattere a ogni simbolo di ingresso e una visualizzazione della conversione del risultato.

Abituatevi a scollegare l'alimentazione del circuito ogni volta che eseguite qualche modifica su di esso, per evitare di provocare danni sui componenti del laboratorio.

Scollegate sempre l'alimentazione quando lavorate con il laboratorio.





# Divisore per 4

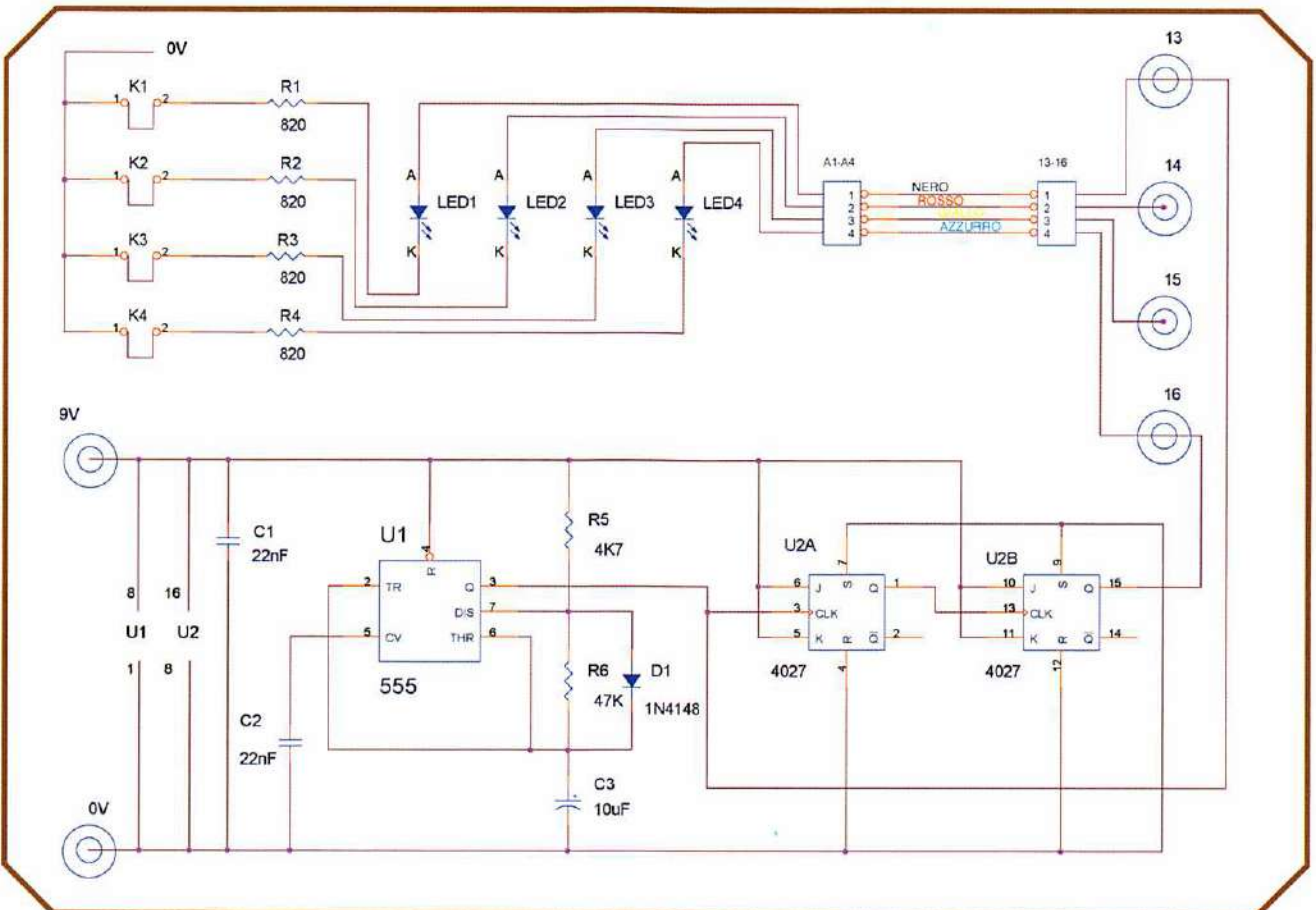
**S**e all'uscita di un generatore si collega l'ingresso di un bistabile, in modo che lo stato di quest'ultimo cambi a ogni impulso di clock, all'uscita del bistabile otterremo un segnale la cui frequenza è la metà di quella di ingresso, ma con un ciclo del 50%, in altre parole, il segnale rimane la stessa quantità di tempo a livello basso e a livello alto. Se disponiamo di due bistabili di questo tipo in cascata otteniamo un divisore per 4.

## Il circuito

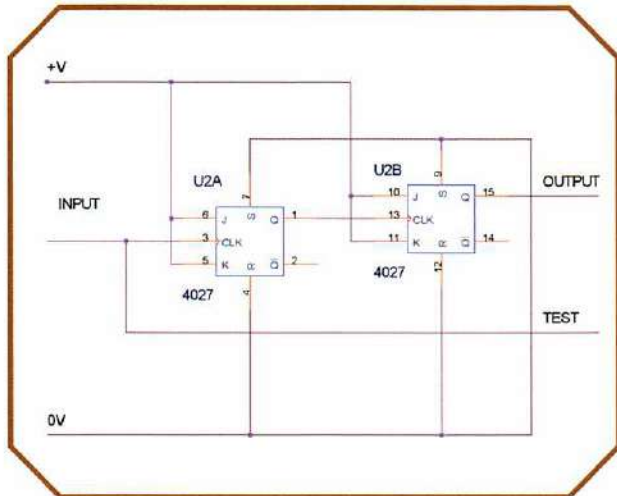
Osservando lo schema possiamo vedere i due circuiti bistabili JK del circuito integrato 4027 collegati in cascata, cioè l'uscita di uno collegata all'ingresso dell'altro. Gli ingressi asincroni che non sono utilizzati e che vengono collegati al negativo dell'alimentazione, corrispondono ai terminali 4, 7, 9, e 12. Il LED 4 si utilizza per visualizzare lo stato dell'uscita.

Il circuito integrato 555 si utilizza per generare un breve impulso, che si ottiene utilizzando il diodo D1, in modo che la carica del condensatore C3 sia rapida. Infatti dobbiamo tenere presente che l'uscita del 555 è a livello alto quando il condensatore si carica, e a livello basso mentre si scarica; la scarica può avvenire solamente attraverso la resistenza R6.

La durata dell'impulso dipende quindi dai valori scelti per R5 e C3, mentre il tempo in cui



Divisore per quattro con il suo generatore di prova.

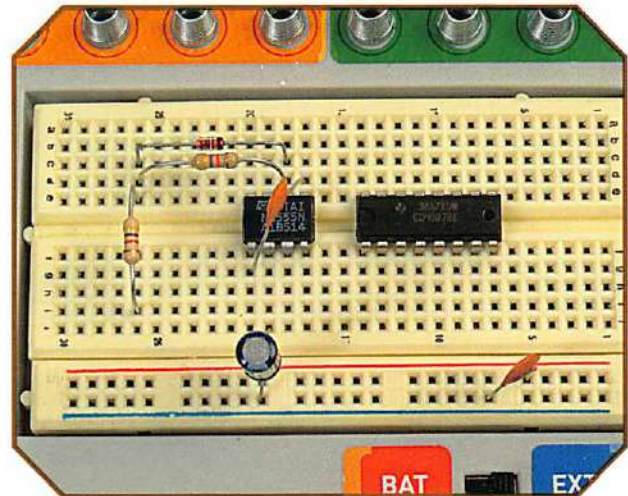


Circuito del divisore per quattro.

l'uscita è a livello basso dipende dai valori di R6 e C3. Il LED 1 si illumina brevemente quando l'uscita del 555, terminale 3, è a livello alto.

## Montaggio

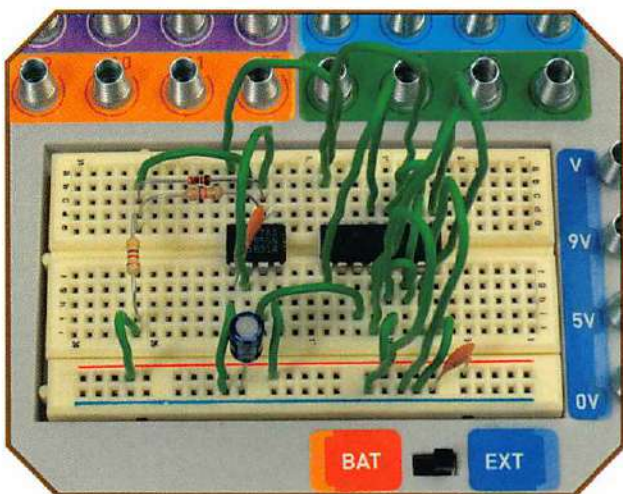
Il montaggio inizia posizionando i componenti sulla scheda Bread Board, facendo particolare attenzione all'orientamento dei circuiti integrati, del diodo D1 e del condensatore C3, perché sono componenti che hanno polarità. In seguito si realizza il cablaggio, iniziando dai fili che stabiliscono i collegamenti all'interno della scheda, senza dimenticare le alimentazioni dei circuiti integrati, e conti-



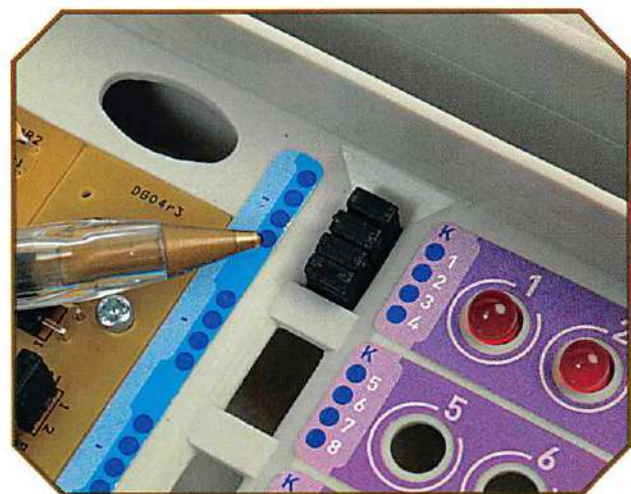
Componenti montati sulla scheda Bread Board.

nuando poi con i fili che vanno alle molle.

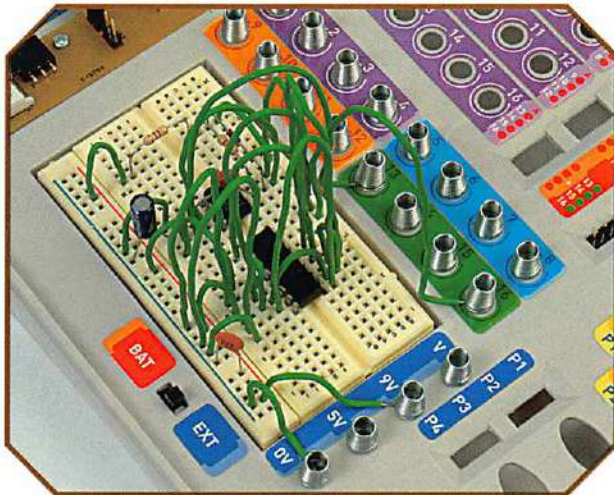
Utilizzeremo anche un cavetto a quattro fili terminato su connettori, fra gli anodi dei LED dal numero 1 al numero 4 e i connettori che portano i segnali alle molle numerate da 13 a 16. È necessario collegare almeno due ponticelli di alimentazione fra i terminali della matrice dei LED indicati come K1 e K2 e il negativo dell'alimentazione che troviamo sui terminali a fianco. Lasciamo scollegati il positivo dell'alimentazione che potrà essere da 9 oppure da 5 V. Questo filo verrà collegato dopo aver verificato che tutto il lavoro di preparazione dell'esperienza sia stato eseguito correttamente.



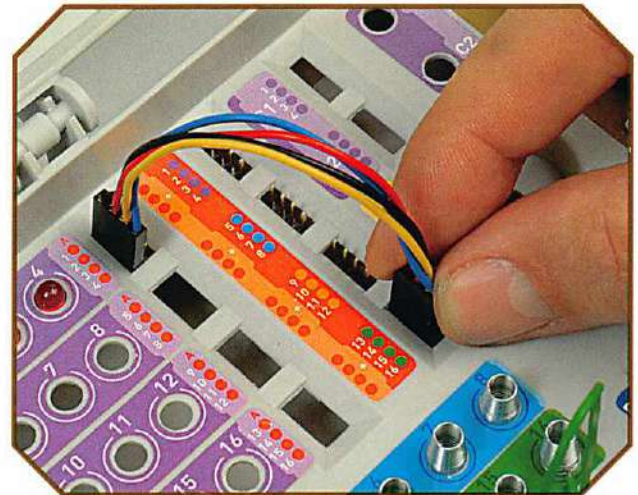
Cablaggio interno.



I ponticelli vanno montati sui catodi dei LED utilizzati.



Collegamento alle molle e all'alimentazione.



Collegamento fra i LED e i connettori delle molle.

## Funzionamento

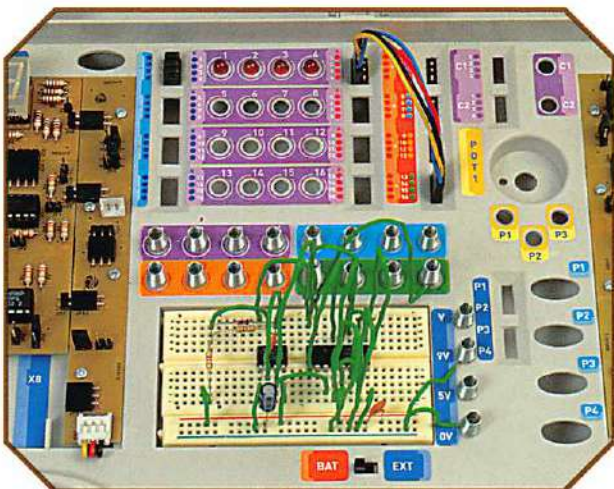
Quando si collega l'alimentazione, il 555 emette degli impulsi di clock che arrivano all'ingresso del primo bistabile T, che si ottiene unendo i due ingressi J e K e collegandoli al positivo, la configurazione genera il cambio di stato dell'uscita a ogni fronte di salita dell'impulso, cioè quando questo passa da livello basso a livello alto.

L'uscita del primo bistabile, terminale 1 di U2 arriva all'ingresso del clock del secondo bistabile, terminale 13 di U2. Osservate che il segnale di uscita il cui stato è visualizzato sul LED 16, rimane per un tempo ugua-

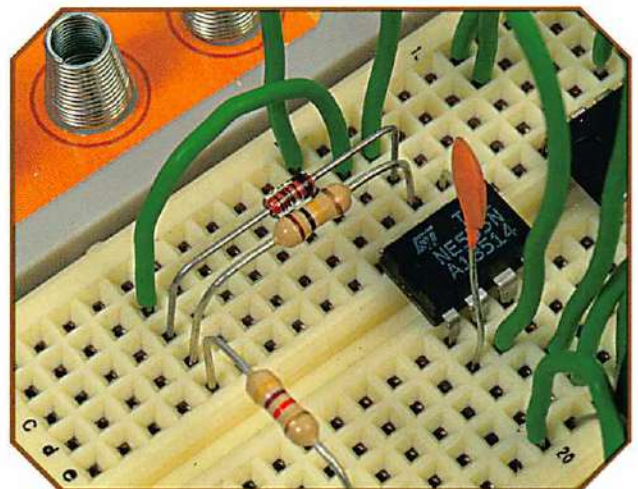
le sia a livello basso che a livello alto, dato che il suo stato cambia al quarto impulso di clock.

## Modifiche

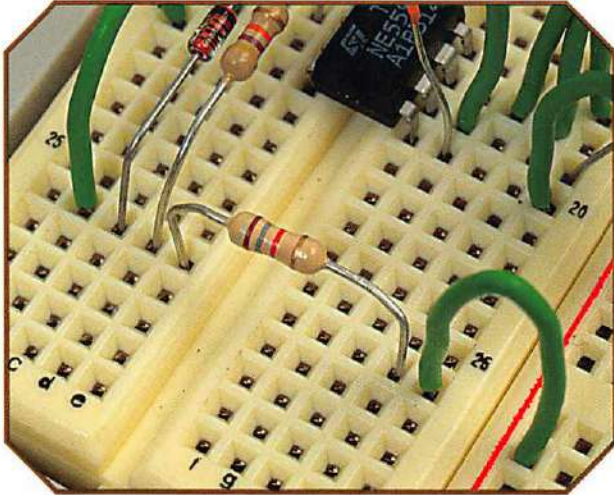
Vi consigliamo di fare delle prove apportando alcune modifiche al circuito, ad esempio sostituendo la resistenza R5 da 4K7 con una da 1K8, questo farà diminuire la durata dell'impulso ma il segnale di uscita continuerà a essere simmetrico; se si toglie il diodo D1 aumenta la durata dell'impulso dato che la corrente di carica del condensatore deve ora attraversare le resistenze R5 e R6. Se si vuole vi-



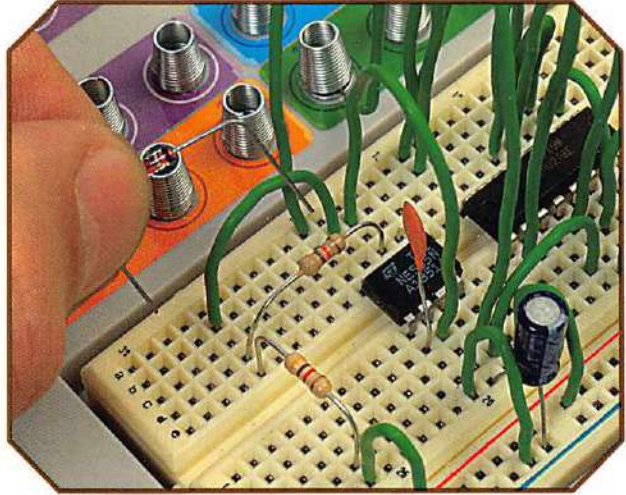
Cablaggio completato.



Aumentando R6 si separano gli impulsi.



Il valore di R5  
fissa la durata dell'intermittenza.



Togliendo D1  
il tempo in cui il LED 1 è illuminato è maggiore.

#### LISTA DEI COMPONENTI

U1	Circuito integrato 555
U2	Circuito integrato 4027
R5	Resistenza 4K7 (giallo, viola, rosso)
R6	Resistenza 47 K (giallo, viola, arancione)
C1, C2	Condensatore 22 nF
C3	Condensatore 10 $\mu$ F elettrolitico

sualizzare lo stato dell'uscita del secondo bi-stabile è sufficiente collegare con un filo il terminale 1 di U2 alla molla 14, in questo modo si illuminerà il LED 2 quando il terminale sarà a livello alto.



Vista del  
laboratorio con  
l'esperimento  
completato.



# Arbitro elettronico

**In molti casi è difficile determinare fra due giocatori chi ha premuto il tasto per primo, tuttavia questo circuito con due pulsanti facilita questo compito, infatti il primo pulsante azionato farà illuminare il LED corrispondente e impedirà che si illumini l'altro, bloccando il circuito di attivazione del LED.**

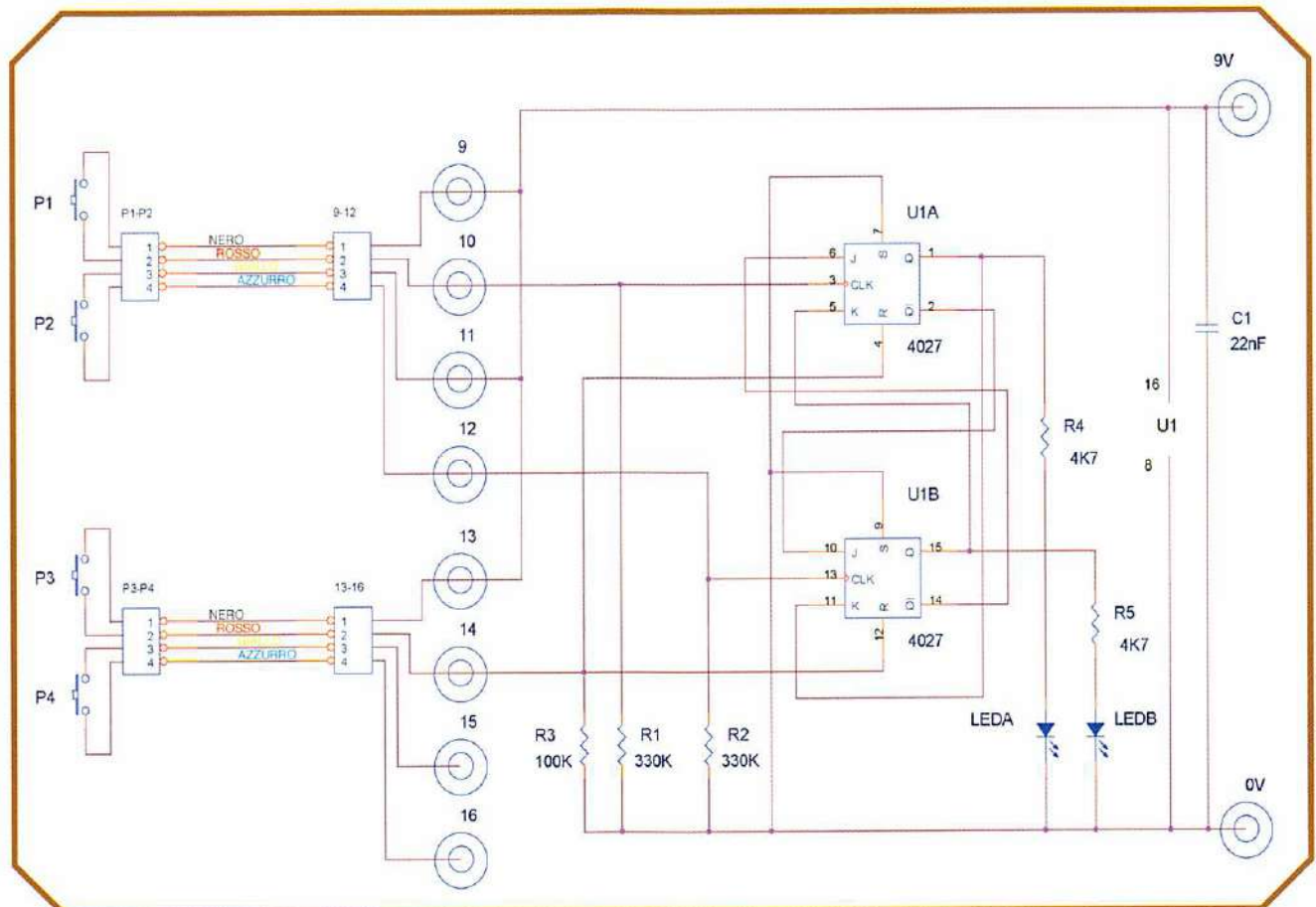
## Il circuito

Se osserviamo lo schema vedremo che i due circuiti bistabili JK del circuito integrato 4027 sono collegati in modo che quando uno dei due riceve un segnale di clock, la sua uscita blocca il funzionamento dell'altro bistabile. Ogni pulsante P1 o P2 applica un fronte di salita all'ingresso del bistabile a cui è collegato. Osservando la figura possiamo vedere che i

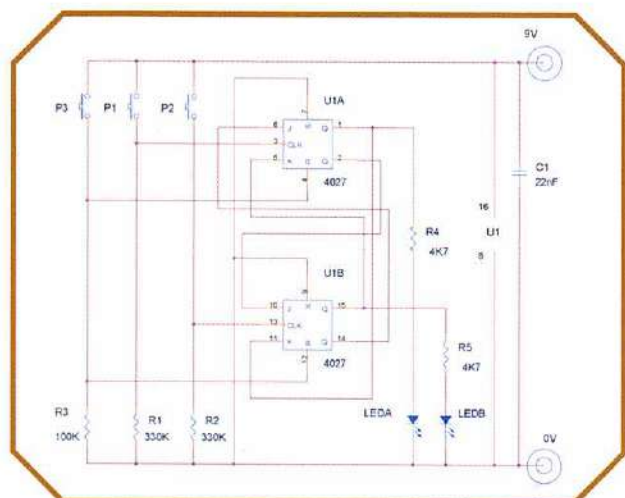
circuiti dei due bistabili sono esattamente uguali fra loro.

Notiamo anche che gli ingressi asimmetrici di entrambi i bistabili sono collegati al negativo dell'alimentazione con una resistenza di pull-down, questo ingresso si può portare a livello 1 premendo P3, ottenendo in questo modo l'azzeramento dell'uscita di entrambi i bistabili.

Il livello di uscita di ogni bistabile, terminali 1



Schema elettrico dell'arbitro elettronico.



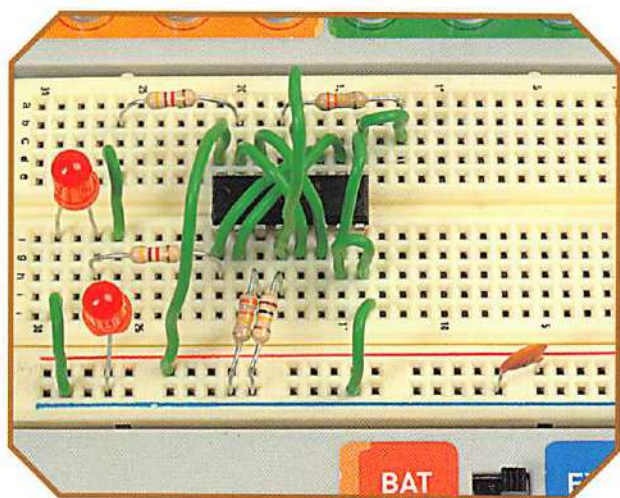
Schema elettrico  
semplificato dell'arbitro elettronico.

e 15, è visualizzato quando è a livello alto, con l'accensione del LED A e LED B rispettivamente.

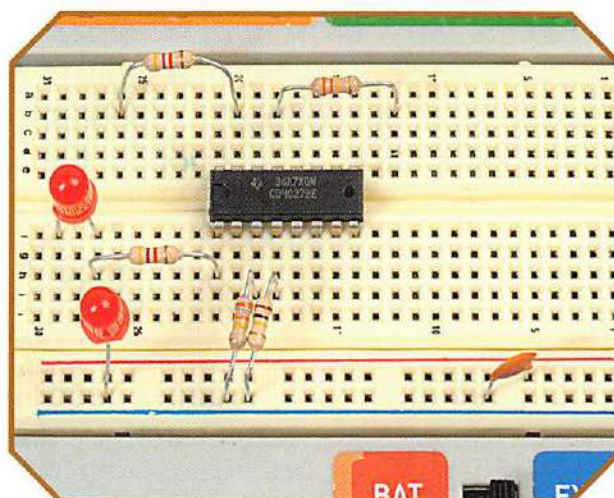
Il condensatore C1 si utilizza come filtro dell'alimentazione.

## Montaggio

Il montaggio inizia dalla collocazione del circuito integrato 4027 sulla scheda Bread Board e prosegue con gli altri componenti, tenendo presente l'orientamento del circuito integrato e la polarità dei LED. Il cablaggio non è difficile, però è piuttosto impegnativo, quindi è facile sbagliare se si cerca di eseguire il lavoro in fretta: è necessario eseguire attentamente lo schema.



Cablaggio interno della scheda.

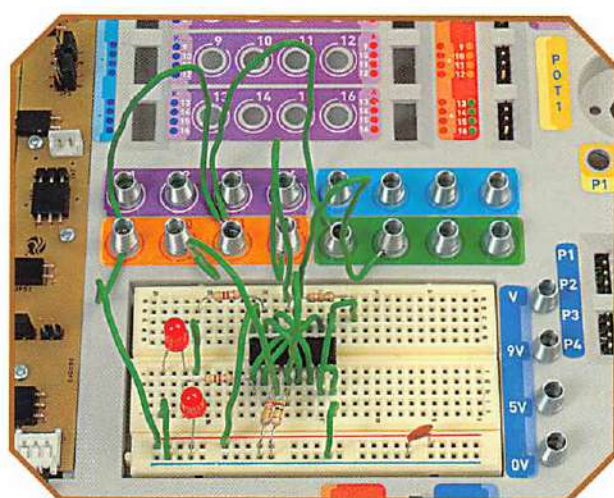


Componenti  
sulla scheda Bread Board.

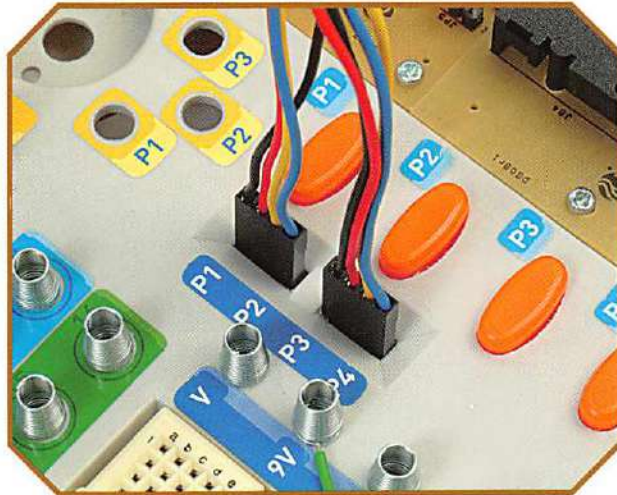
In questo esercizio sono disponibili i quattro pulsanti. I pulsanti P1 e P2 – corrispondenti ai giocatori – si collegano con un cavetto a quattro fili terminato su due connettori. Questo cavetto va inserito sul connettore siglato come P1 P2, situato vicino ai pulsanti, e dall'altro capo al connettore corrispondente alle molle 9-12. In altre parole il collegamento al pulsante P1 sarà disponibile sulle molle 9 e 10 e quello del pulsante P2 sulle molle 11 e 12.

Il terzo pulsante P3, si collega nello stesso modo, utilizzando in questo caso le molle 13 e 14.

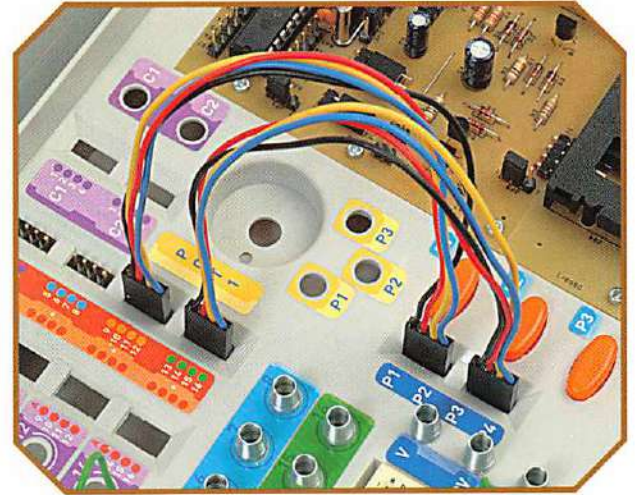
Il negativo dell'alimentazione si collega a 0 V e il positivo a 9 o 5 V, tuttavia, que-



Collegamenti fra la scheda Bread Board e le molle.



Collegamenti dei cavetti alla scheda dei pulsanti.



I collegamenti dei pulsanti sono portati sulle molle dalla 9 alla 14.

st'ultimo collegamento si deve eseguire solo dopo aver verificato che tutti i componenti occupino il loro posto, e tutti i collegamenti siano stati eseguiti in modo corretto.

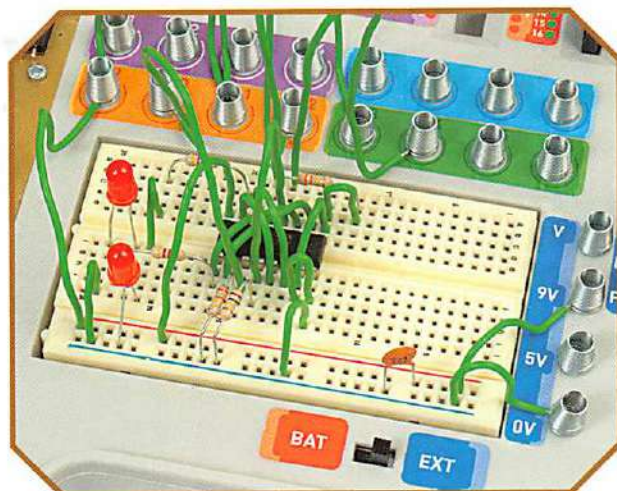
## Funzionamento

Quando si collega l'alimentazione è possibile che uno dei due LED si illumini, tuttavia il gioco inizia premendo P3, in modo da applicare un RESET a entrambi i bistabili, impostando a 0 le uscite e spegnendo i due LED, a questo punto il circuito è pronto per il gioco.

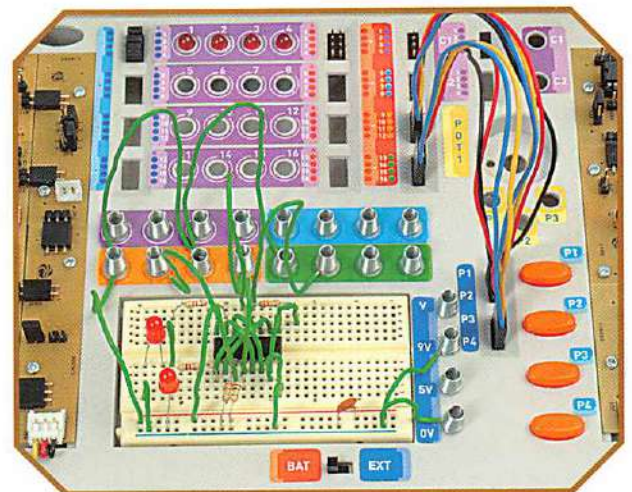
Prima però facciamo ancora una prova: se premiamo P1, il LED A si illumina, se dopo premiamo P2 dobbiamo verificare che il circuito non abbia risposta a questa seconda pulsazione; cioè il LED A continua a rimanere illuminato.

Premendo nuovamente P3 si spegneranno entrambi i LED e se di seguito premeremo P2, il LED B si dovrà illuminare, disattivando P1 in modo che anche se venisse premuto, il LED B continuerà a essere illuminato.

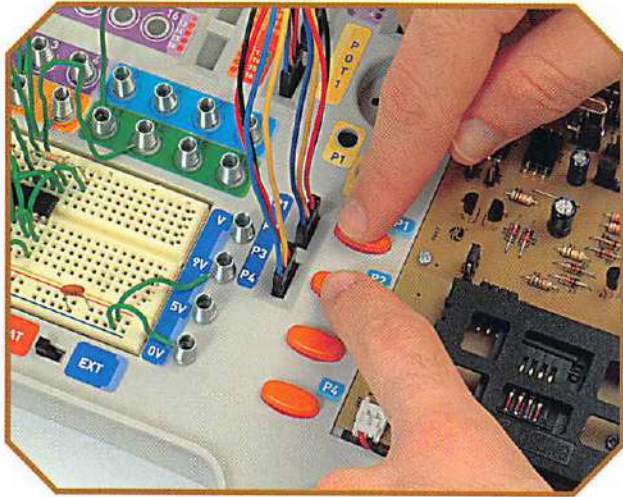
Infine attiveremo P3 per riportare il nostro circuito a essere pronto per la continuazione del gioco.



Grazie al collegamento eseguito con il cavetto a 4 fili possiamo utilizzare i pulsanti sulla scheda Bread Board.

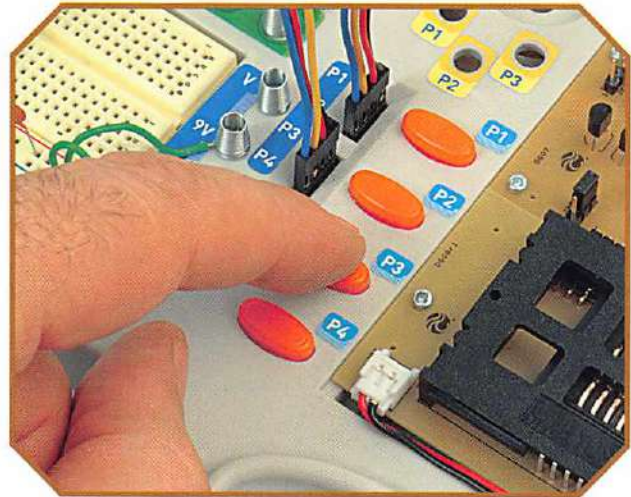


Vista completa del cablaggio.



Anche se si preme contemporaneamente, uno solo sarà il primo.

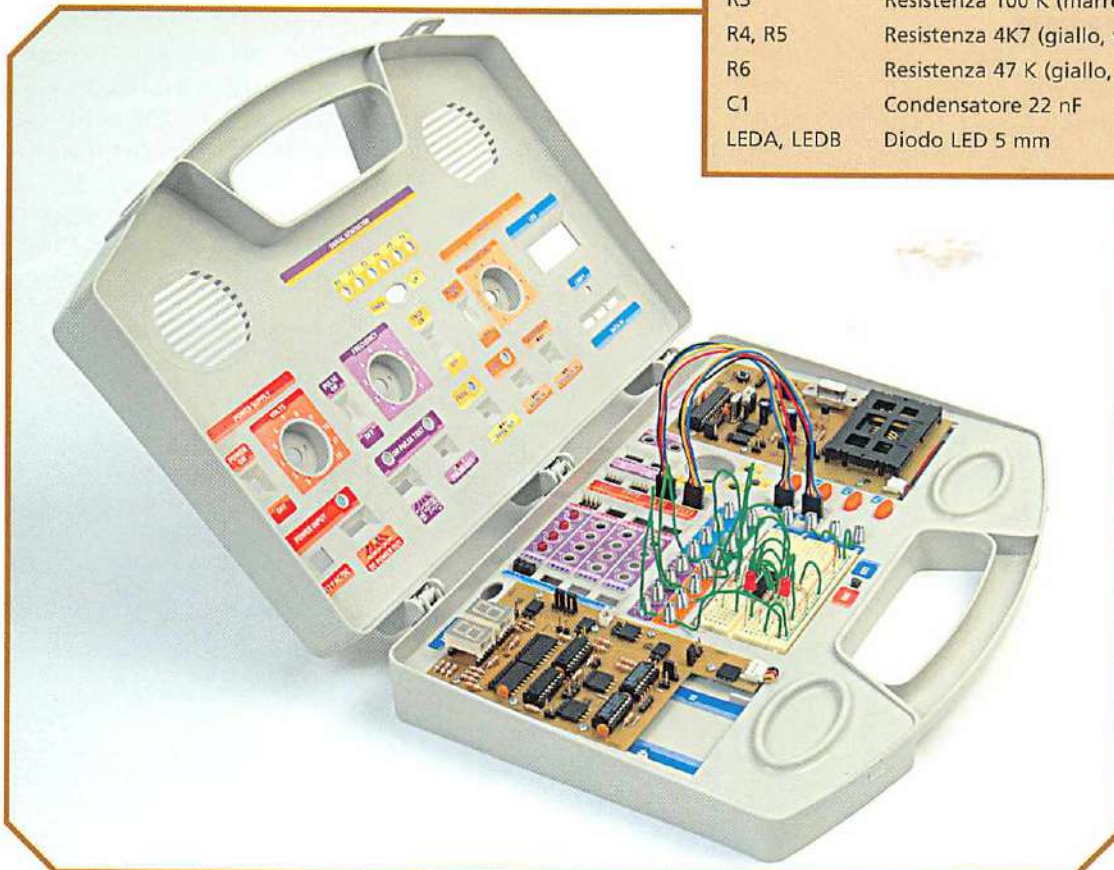
Anche se entrambi i giocatori premeranno contemporaneamente, solo uno risulterà il più veloce, nonostante la differenza potrebbe essere solo di una piccola frazione di secondo.



Premendo P3 si spengono entrambi i LED.

#### LISTA DEI COMPONENTI

U1	Circuito integrato 4027
R1, R2	Resistenza 330 K (arancio, arancio, giallo)
R3	Resistenza 100 K (marrone, nero, giallo)
R4, R5	Resistenza 4K7 (giallo, viola, rosso)
R6	Resistenza 47 K (giallo, viola, arancio)
C1	Condensatore 22 nF
LEDA, LEDB	Diodo LED 5 mm



Vista del laboratorio con l'esperimento completo.





# Indicatore stradale

**Q**uesto circuito simula il movimento, è composto da due file di LED che generano la sensazione di movimento illuminandosi in modo sequenziale, risulta molto utile per richiamare l'attenzione in situazioni di pericolo.

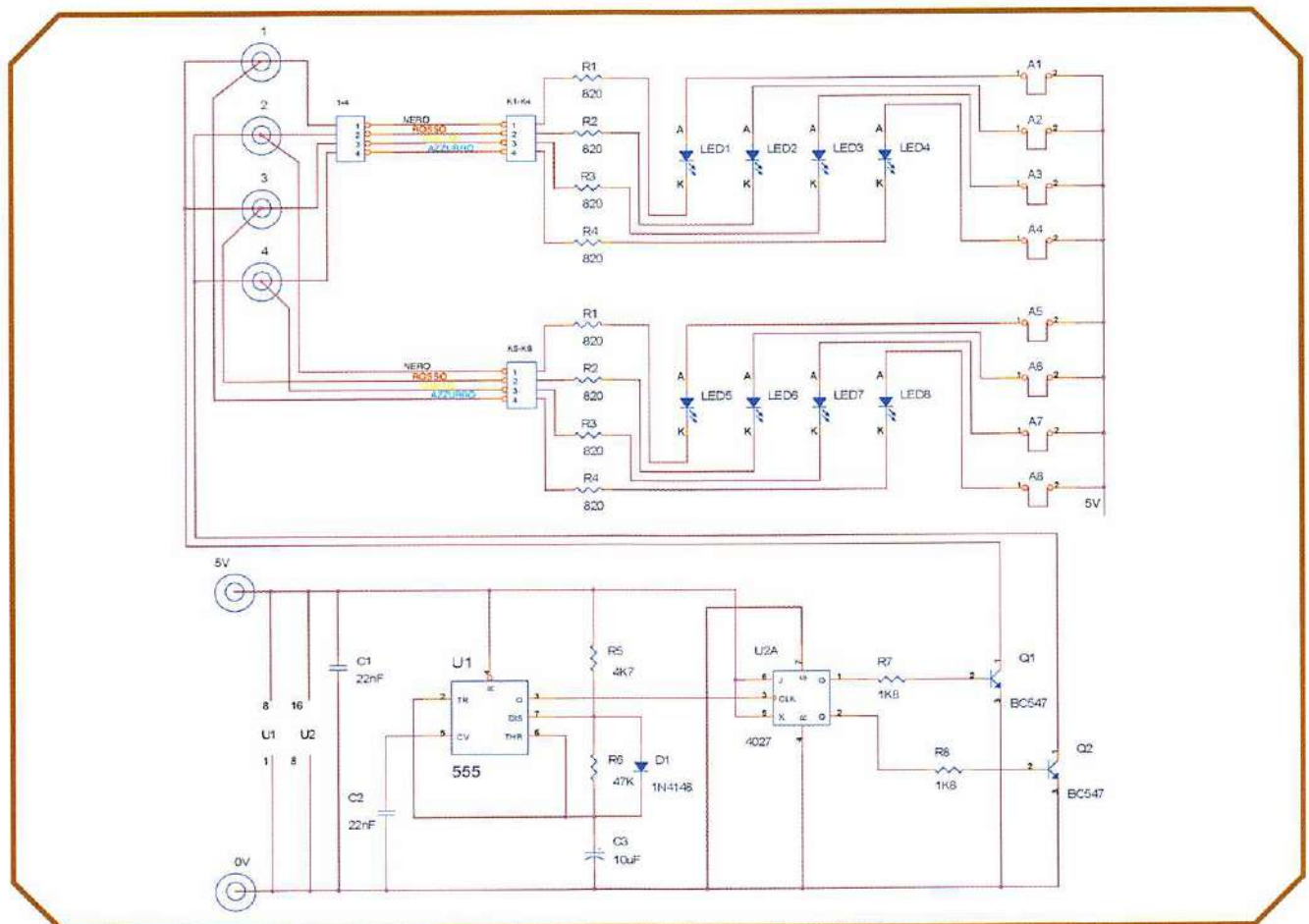
## Il circuito

Osservando il circuito troveremo un astabile formato da un circuito integrato 555 opportunamente configurato, che viene utilizzato per ottenere gli impulsi a una determinata frequenza.

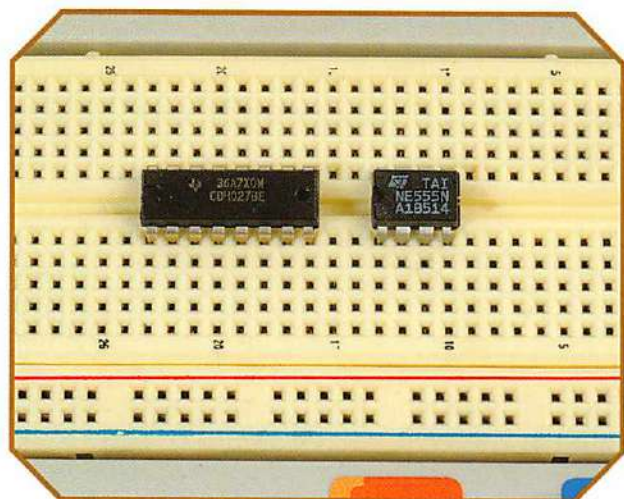
L'intervallo fra gli impulsi si può aumentare incrementando o riducendo la resistenza R6.

L'uscita di questo astabile si applica all'ingresso di uno dei due bistabili contenuti nel circuito integrato 4027, il quale avendo gli ingressi J e K collegati a livello logico 1 funziona come un bistabile del tipo T, quindi lo stato della sua uscita cambia a ogni impulso di clock ricevuto.

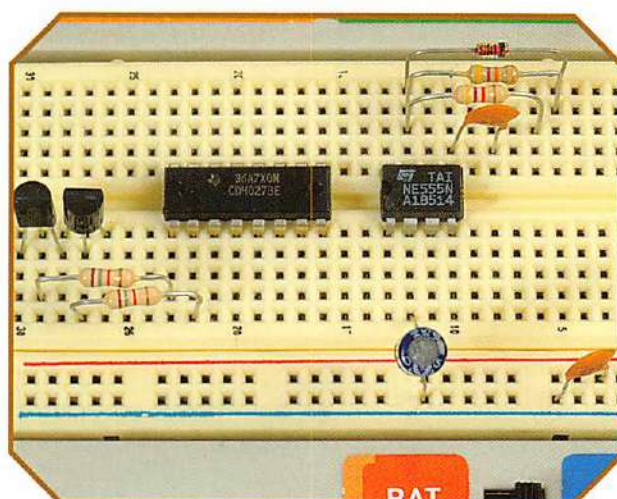
L'uscita Q del bistabile – terminale 1 dell'integrato – si utilizza quando è a livello alto, per attivare il transistor Q1 che entrando



Schema elettrico dell'indicatore luminoso.



I primi componenti da montare sono gli integrati.



Componenti sulla scheda Bread Board.

in conduzione provoca l'accensione dei LED 1, 3, 6 e 8. L'altra uscita del bistabile è opposta e quando l'uscita, terminale 1, passa a 0, l'uscita complementare, terminale 2, passa a livello alto; si spegneranno quindi i LED precedenti e si illumineranno i LED 2, 4, 5 e 7 grazie alla polarizzazione del transistor Q2.

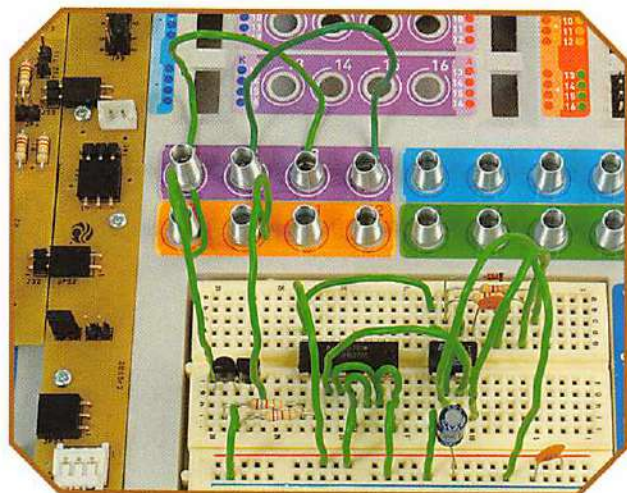
I transistor Q1 e Q2 permettono di ottenere una corrente superiore a quella che potrebbe essere fornita direttamente da una delle porte di questo tipo di integrati, tramite i transistor quindi si evita di sovraccaricare le uscite dell'integrato stesso.

## Montaggio

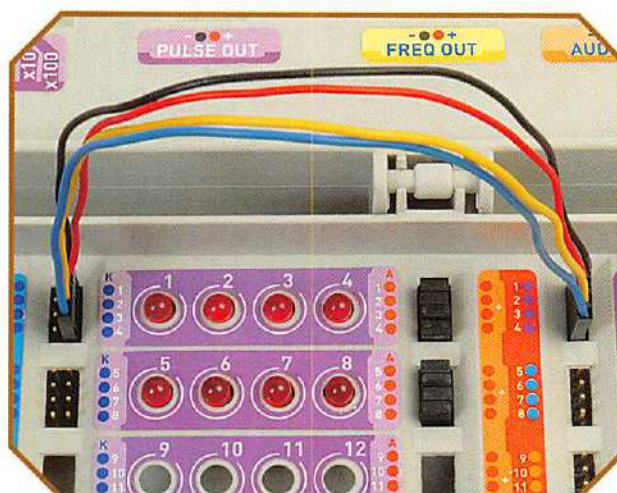
Il montaggio inizia dal posizionamento dei circuiti integrati 555 e 4027 sulla scheda Bread Board e continua facendo particolare attenzione alla polarità del condensatore C3, a quella del diodo e al posizionamento dei terminali dei transistor.

Dato che abbiamo già a disposizione due file di LED, le utilizzeremo per ottenere un maggiore effetto visivo.

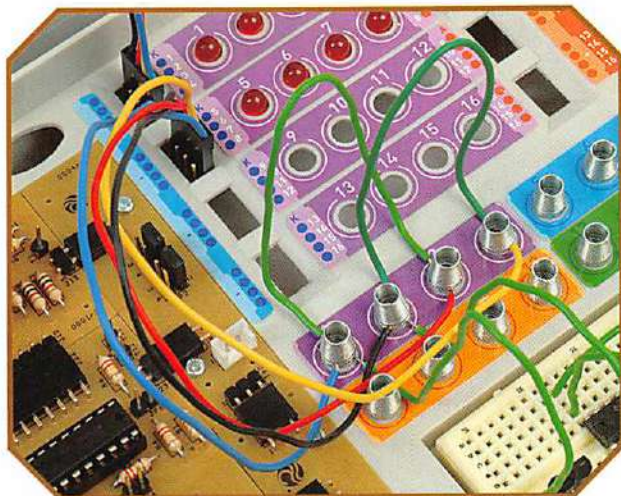
I primi quattro LED si collegano utilizzando un cavetto terminato su due connettori a quattro vie che verranno collegati ai cato-



Collegamenti fra le molle e la scheda Bread Board.



Cavetto a due connettori installato.

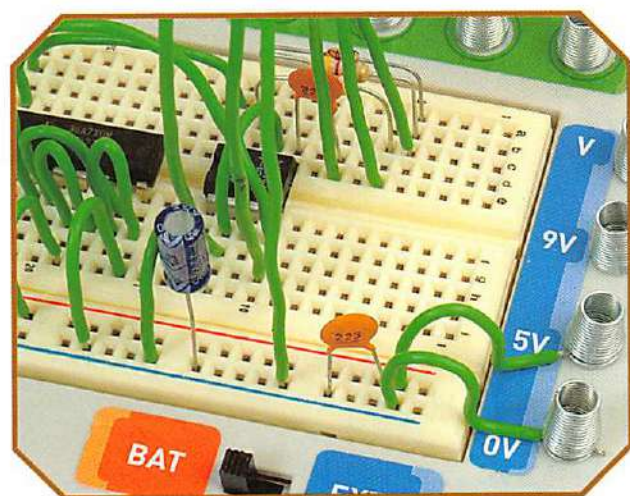


*Cavetto a un connettore installato.*

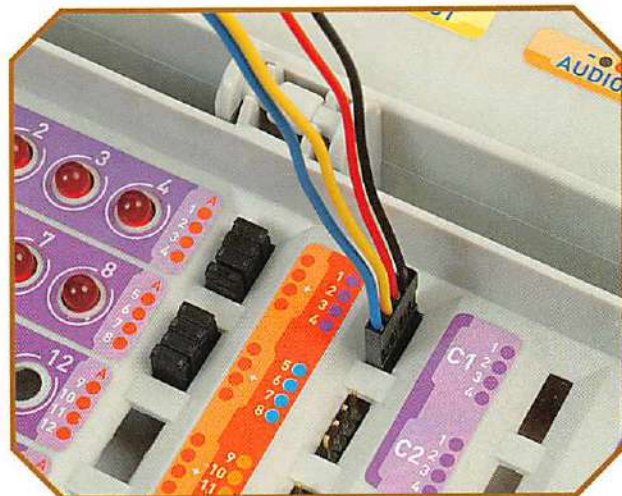
di, terminali da K1 a K4 e alle molle dalla 1 alla 4.

I collegamenti della successiva fila di LED si realizzano con un cavetto terminato su un connettore, e con i fili liberi dall'altro capo. Questo cavetto si collega fra i catodi da K5 a K8 e alle molle precedenti, ma in modo che si illuminino i LED dal numero 5 al numero 8 in ordine inverso, cosa che si ottiene seguendo lo schema.

Per fare in modo che il circuito funzioni bisogna montare gli otto ponticelli corrispondenti agli anodi degli otto LED, per ottenere il collegamento fra questi e il positivo da 5 V.



*I collegamenti a 5 V si eseguono dopo aver verificato il montaggio.*



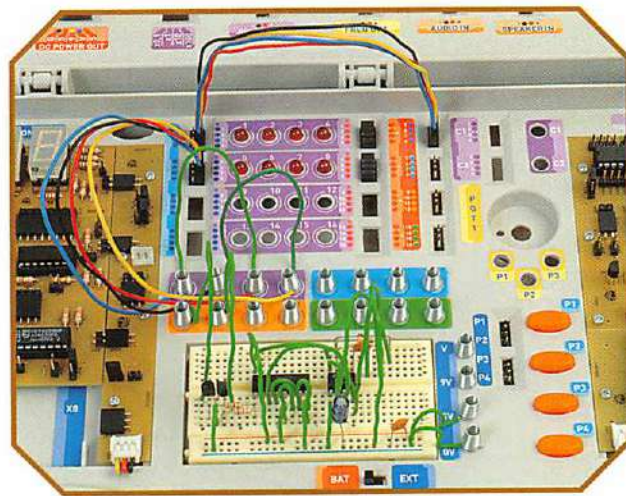
*Dettaglio ingrandito dei ponticelli per gli otto anodi.*

## Alimentazione

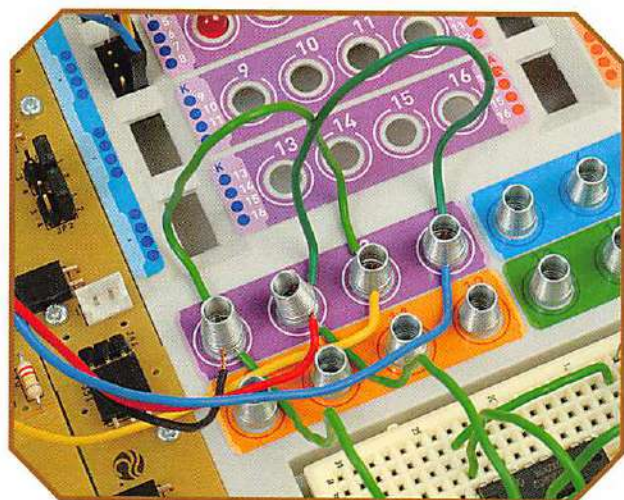
Per far funzionare il circuito bisogna montare gli otto ponticelli corrispondenti agli anodi degli otto LED per ottenere il collegamento fra questi e il positivo da 5 Volt, disponibile sui connettori vicini agli anodi.

Il resto del circuito si deve alimentare quindi a 5 Volt, utilizzando la molla di 0 V e quella di 5 V, lasciando questi collegamenti per ultimi, quando tutto il lavoro sarà stato terminato e controllato.

È necessario inoltre verificare di aver collegato le alimentazioni ai circuiti integrati, che



*Vista completa del cablaggio realizzato.*

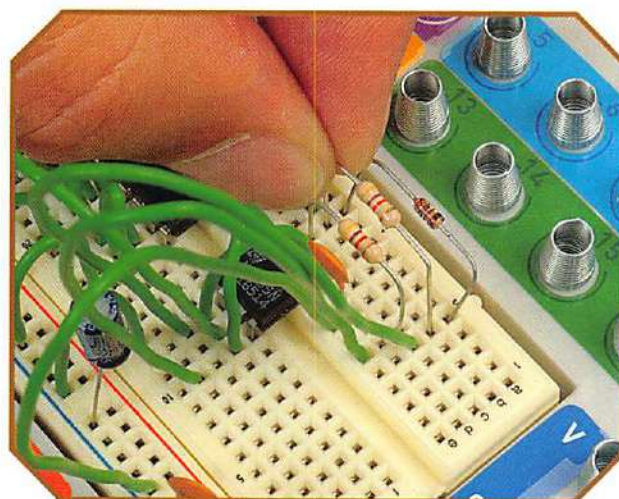


*Cambiando i collegamenti dei catodi si ottiene un altro effetto.*

per il 555 sono le seguenti: terminale 8 positivo e 1 negativo. Per il 4027 invece sono: terminale 16 positivo e 8 negativo.

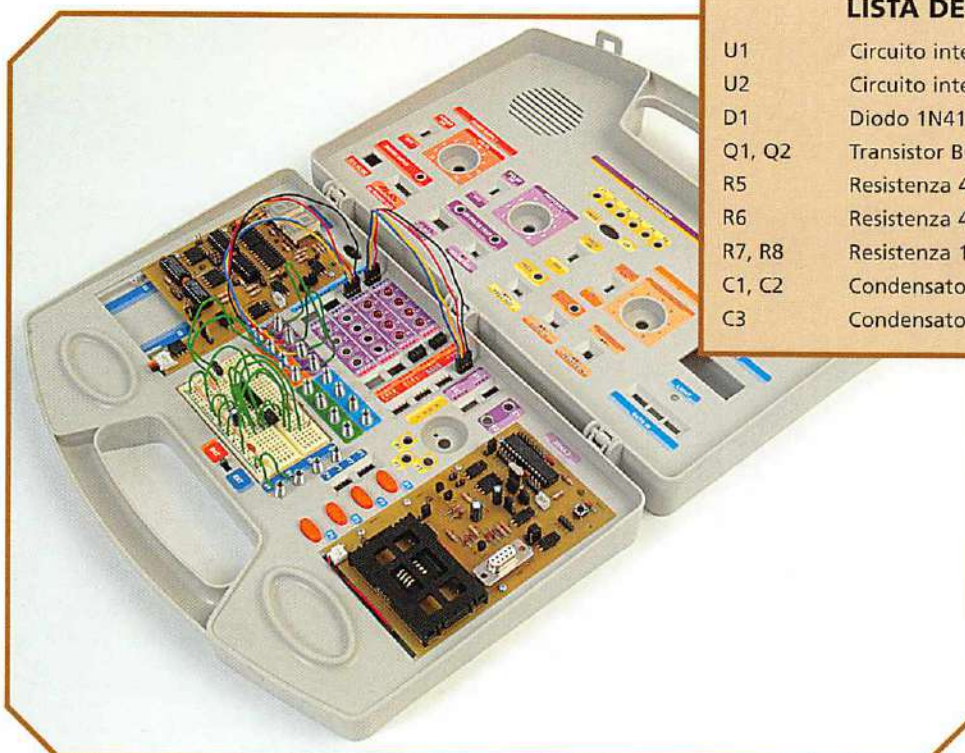
## Funzionamento

Il circuito deve funzionare appena si collega l'alimentazione, anche se la frequenza è bas-



*Diminuendo R6 a 4K7 si ottiene un buon effetto luminoso.*

sa per la nostra applicazione, la possiamo aumentare o diminuire tramite il valore della resistenza R8; possiamo provare con 4K7 e con 1K8. Possiamo anche realizzare diverse combinazioni fra i collegamenti dei catodi dei LED, per ottenere effetti luminosi diversi.



### LISTA DEI COMPONENTI

U1	Circuito integrato 555
U2	Circuito integrato 4027
D1	Diodo 1N4148
Q1, Q2	Transistor BC547 o BC548
R5	Resistenza 4K7 (giallo, viola, rosso)
R6	Resistenza 47 K (giallo, viola, arancio)
R7, R8	Resistenza 1K8 (marrone, grigio, rosso)
C1, C2	Condensatore 22 nF
C3	Condensatore 10 $\mu$ F elettrolitico

*Vista del laboratorio con l'esperimento completato.*



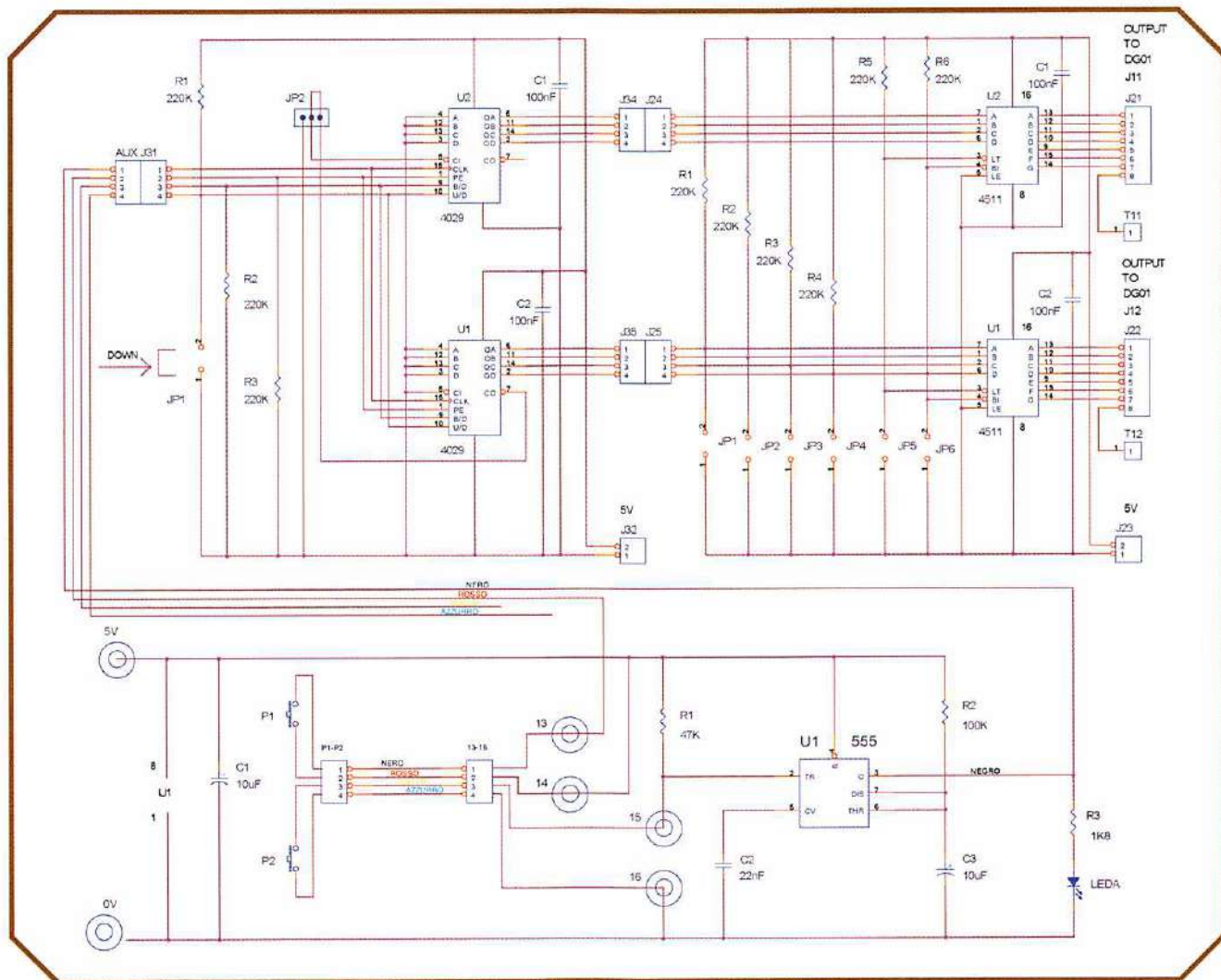
# Conta-impulsi

**A** prima vista può apparire facile utilizzare un pulsante per applicare un singolo impulso all'ingresso di un contatore, ma l'apparenza inganna. All'inizio una resistenza di pull-down e un pulsante sembrano sufficienti, ma questo sistema può dare molti problemi e più di un grattacapo, in particolare per chi è ancora alle prime armi in questa difficile, ma appassionante disciplina dell'elettronica.

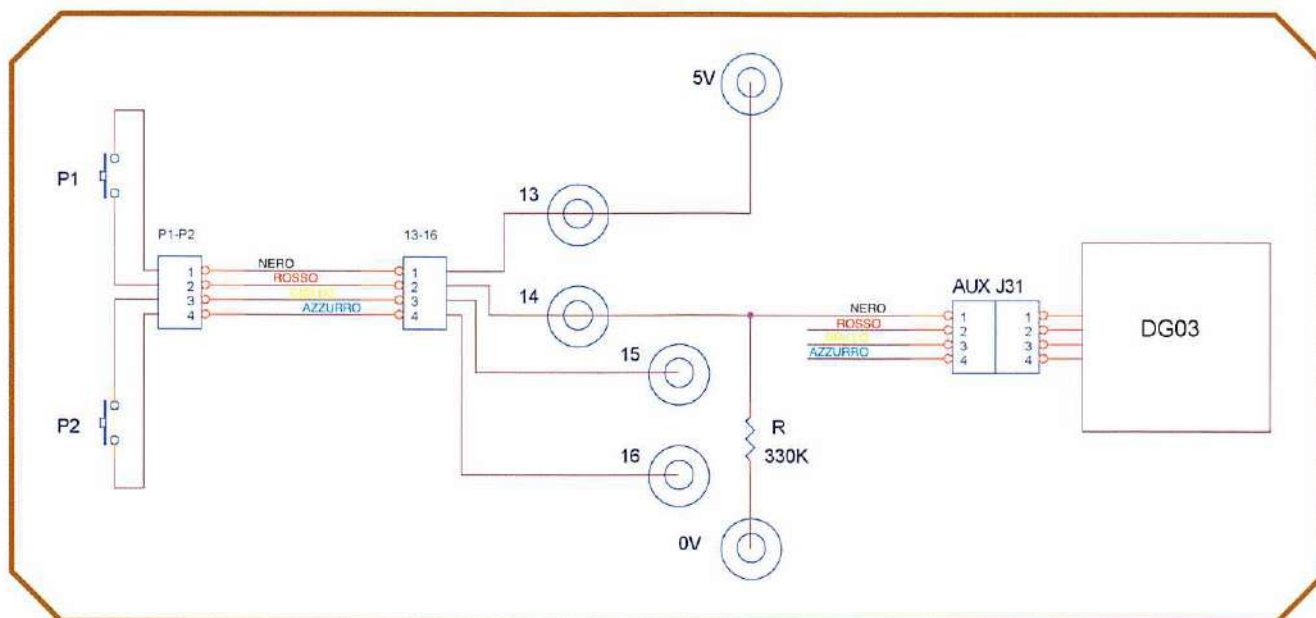
## Il problema

Prima di iniziare l'esperimento vi consigliamo di montare e utilizzare il semplice circuito con un pulsante e una resistenza di pull-down, per verificare quanto sia difficile generare un sin-

golo impulso in questo sistema. Potrete osservare che il contatore avanza di diverse unità e in modo quasi imprevedibile, dato che rileva diversi impulsi al suo ingresso, tutti considerati validi, il conteggio quindi rappresenterà il numero di impulsi rilevati.



Schema del contatore di impulsi.



*Il collegamento diretto dei pulsanti può creare problemi.*

Dopo aver messo a fuoco il problema vedremo alcune soluzioni e, pur avendo già trattato alcuni circuiti che superano questa difficoltà, non vorremmo che qualche lettore coltivasse la speranza che un solo pulsante possa bastare a porre rimedio ai problemi incontrati quando utilizzavamo un filo al posto del pulsante stesso.

## Il circuito

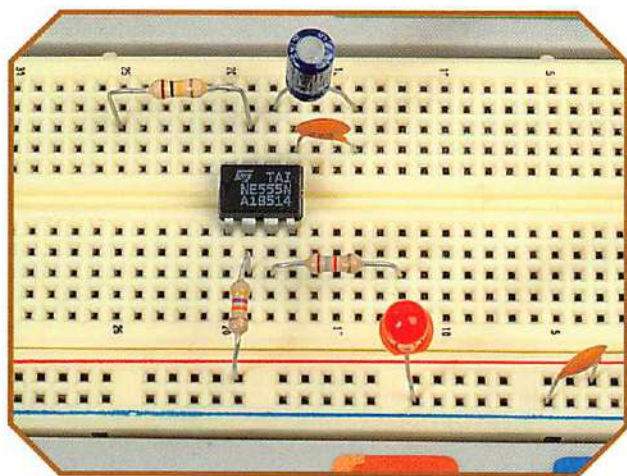
Se osserviamo lo schema potremo vedere che è molto esteso, dato che vi è rappresentato

tutto il contatore. Per ora prenderemo in considerazione la parte inferiore.

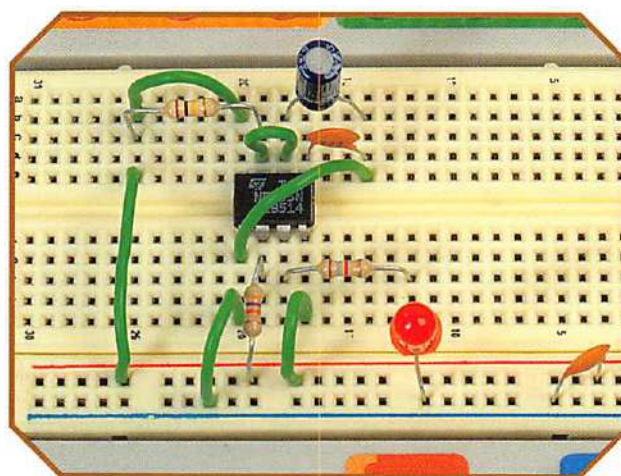
Il pulsante P2 si utilizza per applicare impulsi al contatore, però non sono applicati direttamente ma è stato inserito un circuito monostabile, e sarà l'uscita di quest'ultimo ad applicare un singolo impulso all'ingresso del contatore.

Il circuito monostabile si realizza utilizzando un 555 configurato allo scopo. Sulla sua uscita si collega un LED per visualizzare la durata dell'impulso di uscita.

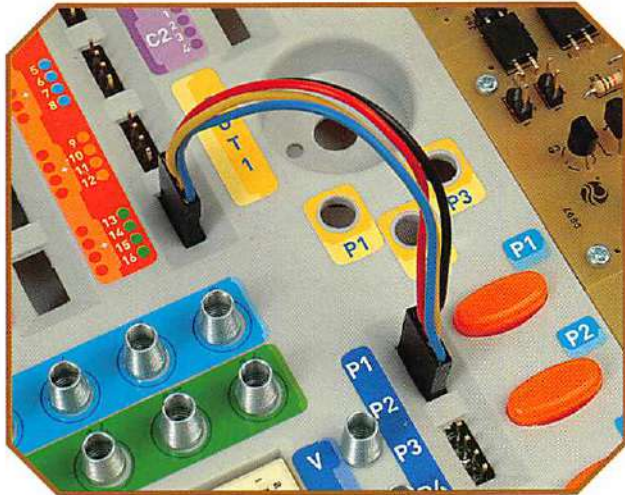
La durata di questo impulso di uscita si può



*Componenti installati sulla scheda Bread Board.*



*Cablaggio interno.*



Collegamento del cavetto ai pulsanti.

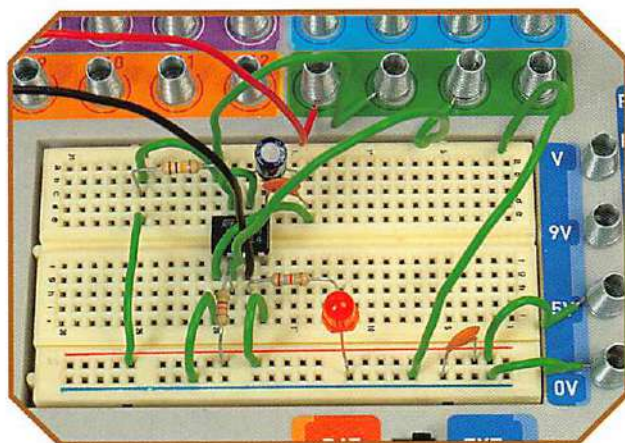
cambiare modificando i valori dei componenti R2 e C3.

Il pulsante P1 si utilizza per azzerare il contatore.

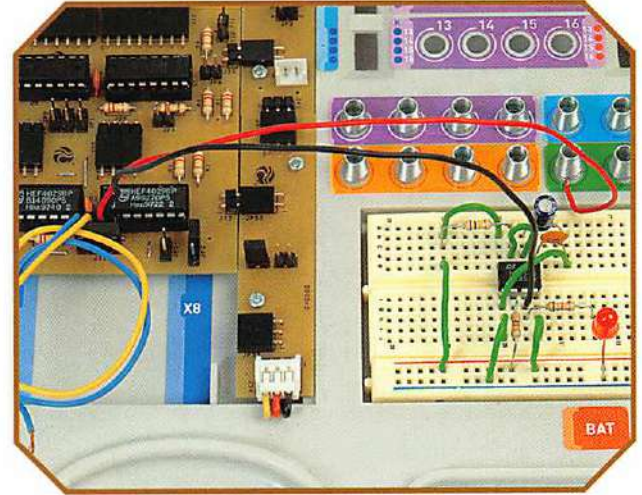
## Montaggio

Il montaggio inizia, come d'abitudine, dal posizionamento del circuito integrato 555 e dai componenti montati sulla scheda Bread Board, realizzando i ponticelli interni alla scheda stessa con dei fili e i collegamenti alle molle.

Il collegamento ai pulsanti si realizza con l'utilizzo di un cavetto terminato su due connettori a quattro vie, collegato tra il connettore identificato come P1 P2 e quello corrispondente alle molle dalla 13 alla 16.



Collegamento alle molle di alimentazione.



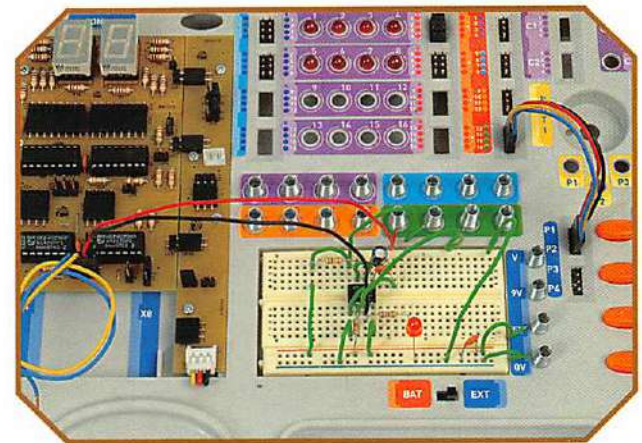
Collegamento del cavetto al contatore.

I collegamenti all'ingresso del clock del contatore e al reset dello stesso si eseguono con un cavetto a quattro fili terminato su un connettore a quattro vie.

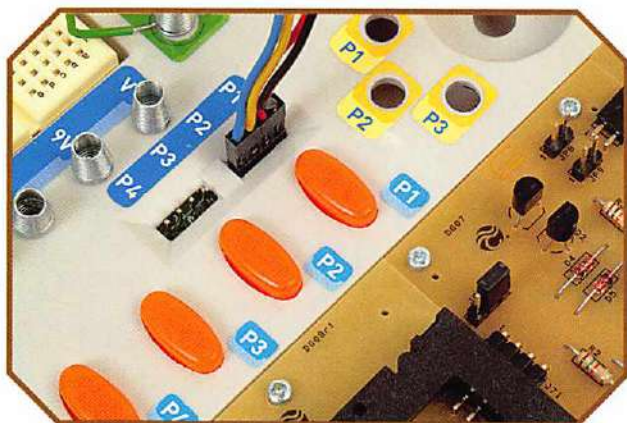
Collegandolo a J31 della scheda DG03, scheda dei contatori, in modo che il filo nero corrisponda al numero 1, avremo a disposizione il collegamento del clock su questo filo nero e quello del reset sul filo rosso.

## Funzionamento

Ogni volta che si preme P2 il contatore avanza di una unità e bisogna attendere che si spenga il LED per poter generare un nuovo impulso. Dato che si tratta di un circuito sperimentale, questo tempo è molto lungo in modo da poterne osservare il funzionamento, è però pos-



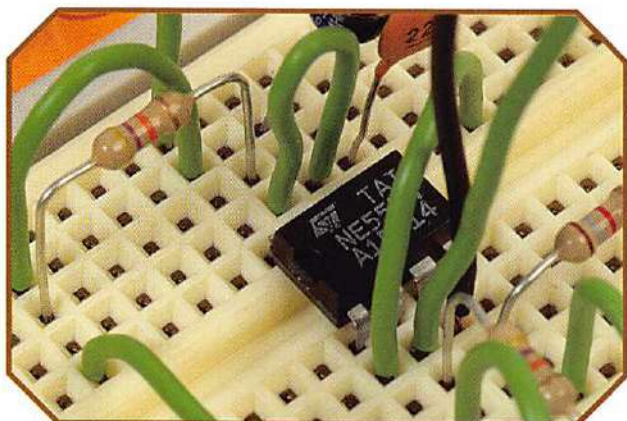
Cablaggio completato.



Premendo P1 si azzerò il contatore.



Ogni volta che si preme P2 il contatore avanza.



Abbassando il valore di R2 si spegne prima il LED.

sibile ridurlo abbassando i valori di R2 e di C3. In questo modo, arriverà all'ingresso del contatore un singolo impulso di clock per il conteggio.

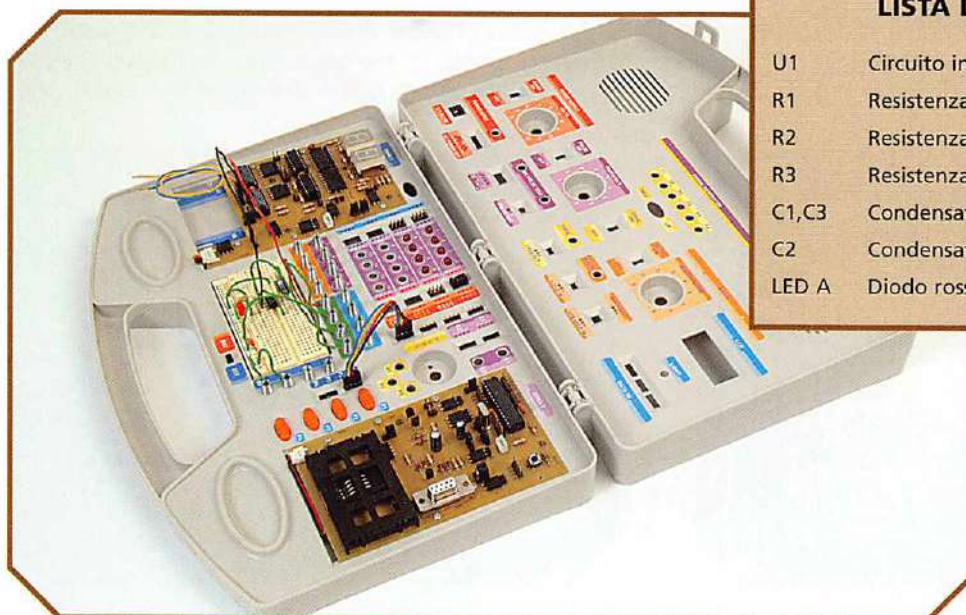
Attivando il pulsante P1 si porta a zero il contatore.

Per ottenere che il contatore conti in senso inverso è necessario collegare il ponticello DOWN sul connettore JP1 della scheda DG03.

Il ponticello JP2 della scheda DG03 deve essere collegato come indicato nello schema, e come si può vedere nelle fotografie, in questo modo si trasmette il riporto e il contatore passa da 09 a 10.

#### LISTA DEI COMPONENTI

U1	Circuito integrato 555
R1	Resistenza 47 K (giallo, viola, arancio)
R2	Resistenza 100 K (marrone, nero, giallo)
R3	Resistenza 1K8 (marrone, grigio, rosso)
C1,C3	Condensatore 10 $\mu$ F elettrolitico
C2	Condensatore 22 nF
LED A	Diodo rosso 5 mm



Vista del laboratorio con l'esperimento completato.





# Generatore di numeri casuali

**Q**uesto circuito si utilizza per ottenere un numero casuale compreso fra 00 e 99. Il numero si legge quando si preme il pulsante, in questo caso P2. Il pulsante P1 si utilizza per verificare che sia possibile generare tutti i numeri e che nessuno venga eliminato.

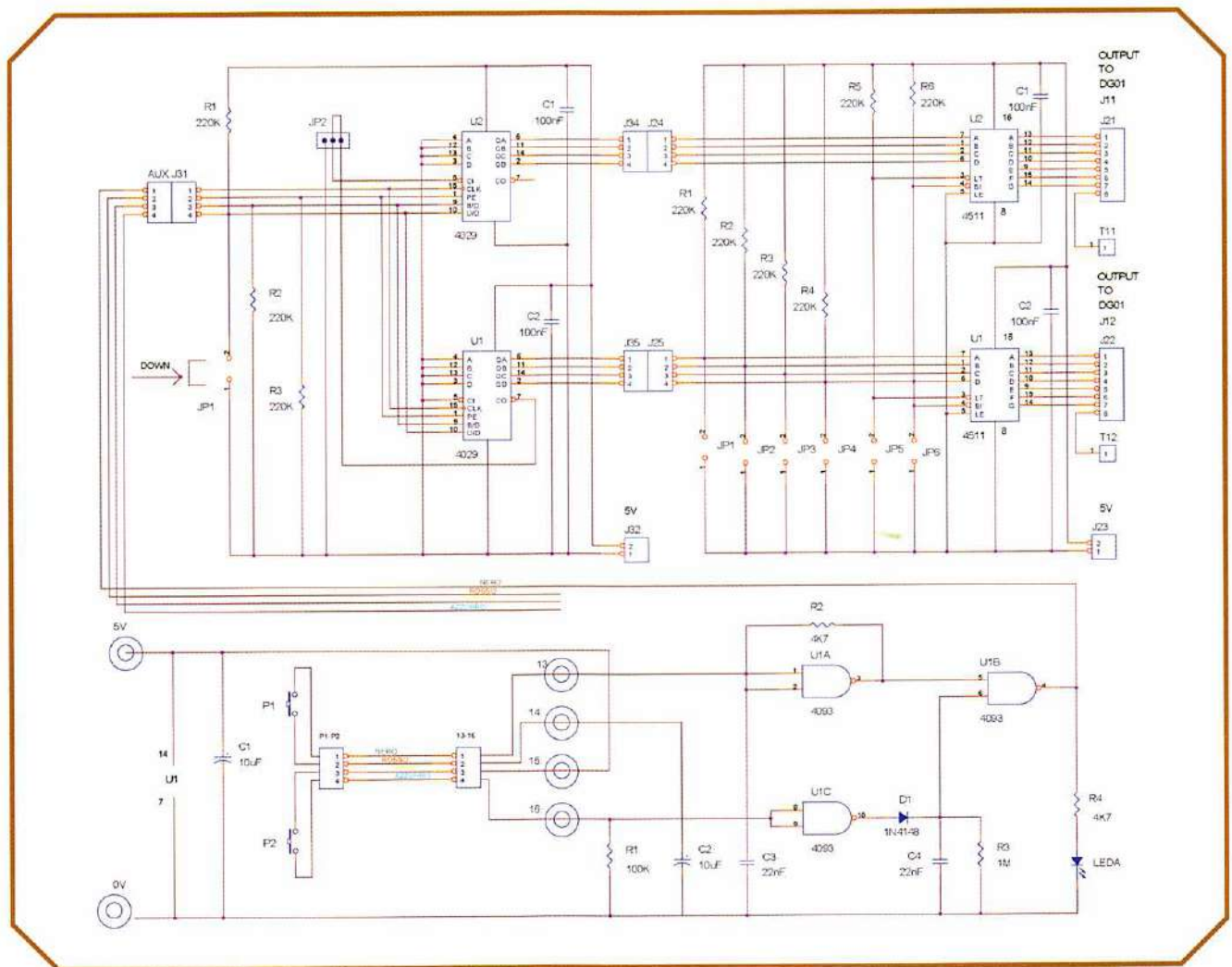
## Il problema

Lo schema è molto grande, dato che sono rappresentate la scheda dei contatori DG03 e quella dei driver DG02, ma il circuito che bisogna aggiungere risulta essere piuttosto semplice.

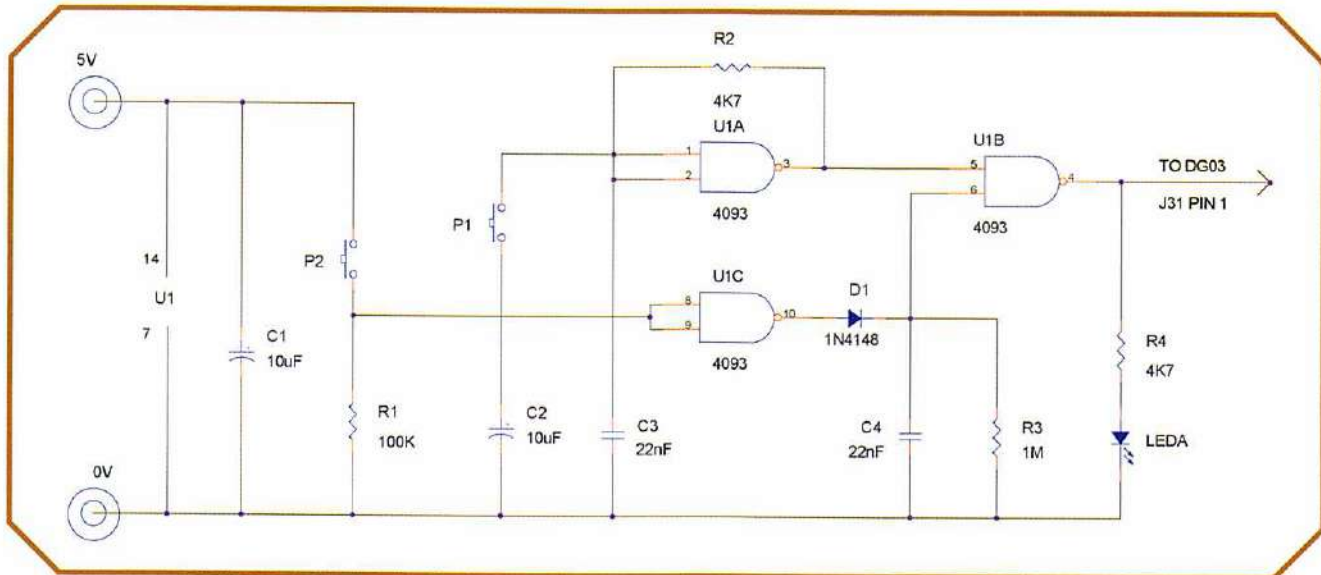
La porta U1A del 4093 si utilizza per ottene-

re un oscillatore astabile, la cui frequenza di oscillazione deve essere sufficientemente alta per impedire all'occhio di recepire le informazioni sullo stato del conteggio del contatore, questa frequenza dipende dal valore della resistenza R2 e dalla capacità del condensatore C3.

La porta U1B si utilizza per applicare o meno il segnale del clock dell'astabile all'ingres-



Schema elettrico del generatore di numeri casuali.



Schema del circuito di controllo.

so del contatore, che corrisponde al terminale 1 del connettore J31 della scheda DG03.

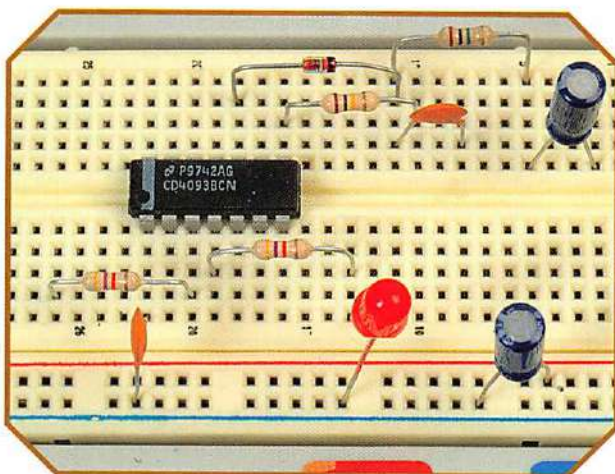
La porta U1C si utilizza per controllare la porta U1B e il suo ingresso in stato di riposo è a livello basso, mantenuto basso per la resistenza R1. Quando l'uscita della porta U1C è a livello alto, la porta U1B lascia passare il segnale del clock verso l'ingresso del contatore; quando questa è a livello basso il condensatore C4 si scarica tramite la resistenza R3 e quando arriva a 0, la porta U1B impedisce che il segnale di clock arrivi all'ingresso del contatore.

Se si preme P2 l'ingresso della porta U1C passa a livello alto e la sua uscita a livello bas-

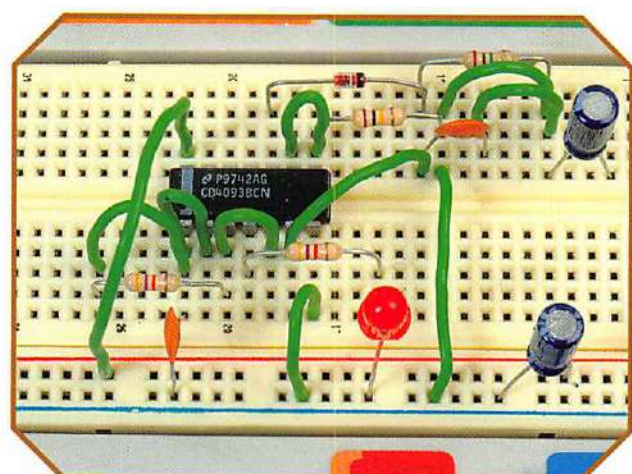
so, il conteggio del contatore si ferma e sarà possibile leggere un numero sul display.

Il pulsante P1 collega un condensatore C2 da 10  $\mu\text{F}$ , in parallelo con C3 da 22 nF, quindi la frequenza dell'oscillatore diventa così bassa da permettere di osservare la frequenza di conteggio del contatore sul display.

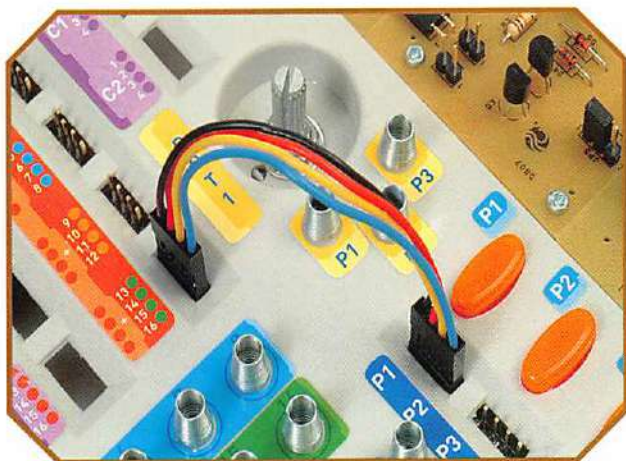
Il diodo D1 e il condensatore C4 generano un certo ritardo del segnale, che rende ancor più difficile la possibile predizione del numero di uscita. La resistenza R3 si utilizza per scaricare il condensatore C4, dato che il consumo della porta è praticamente trascurabile e il condensatore non si scaricherebbe tramite essa.



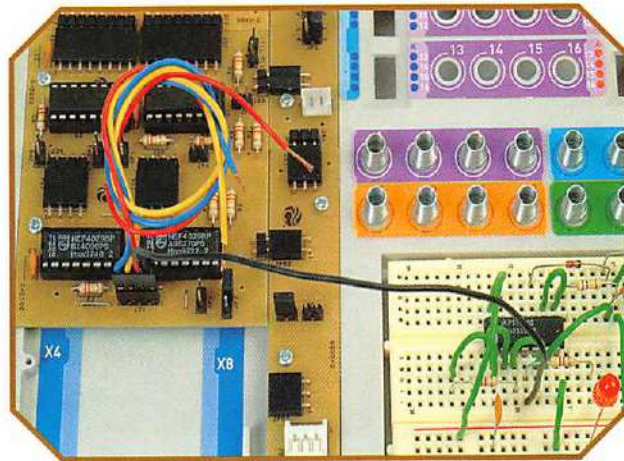
Componenti sulla scheda Bread Board.



Cablaggio interno della scheda.



Collegamento dei pulsanti.



Collegamenti agli ingressi del clock del contatore.

## Montaggio

Il montaggio inizia dal posizionamento del circuito integrato 4093 e degli altri componenti sulla scheda Bread Board, tenendo presente l'orientamento del circuito integrato e la polarità dei diodi, sia del diodo di segnale D1 che del LED.

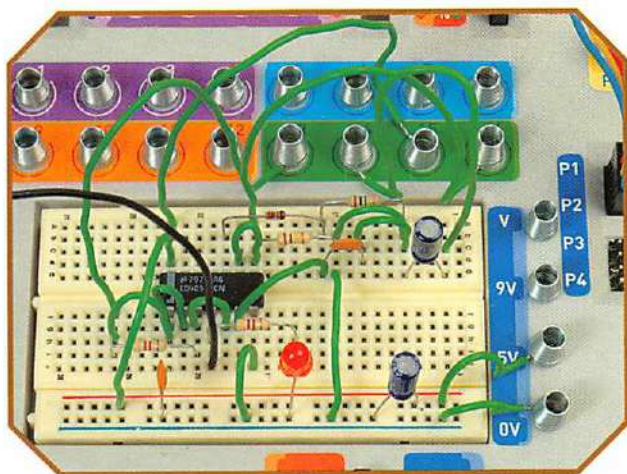
Il cablaggio ai pulsanti si esegue con un cavetto terminato su due connettori, che verranno utilizzati nella zona siglata come P1 P2 posizionata vicino ai pulsanti, e sul connettore corrispondente alle molle 13-16; quindi i collegamenti dei pulsanti P1 e P2 saranno disponibili sulle molle dalla 13 alla 16.

I collegamenti si eseguono con del filo, e deve essere lasciato per ultimo il collegamento

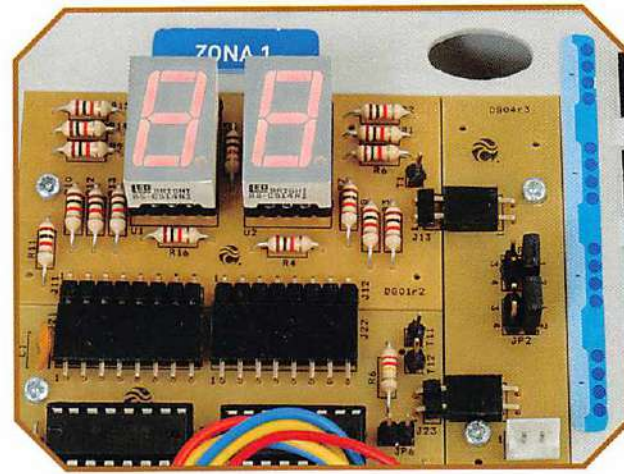
al positivo dell'alimentazione, che in questo caso può essere da 5 o 9 V, tenendo però presente che sia la scheda del contatore sia quella della parte del circuito montata sulla scheda Bread Board, dovranno utilizzare la stessa tensione di alimentazione.

## Funzionamento

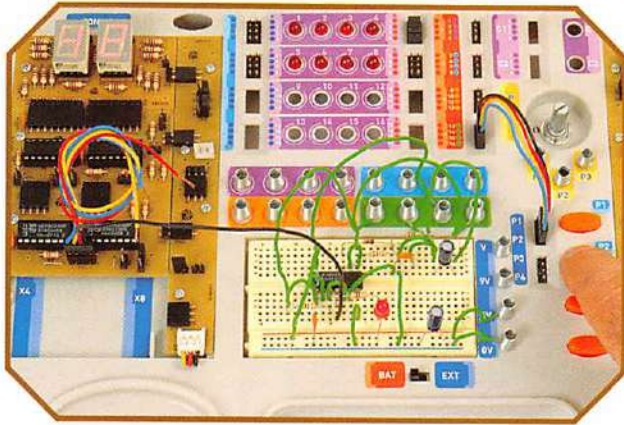
Dopo aver montato e verificato tutto il lavoro svolto si potrà collegare l'alimentazione. Eseguendo questo collegamento, i sette segmenti del display doppio del contatore e il LED A si illumineranno. In realtà il contatore starà contando da 00 a 99, ma in modo così veloce che l'occhio umano non può percepire quale numero sta passando. Premendo P1 il contatore



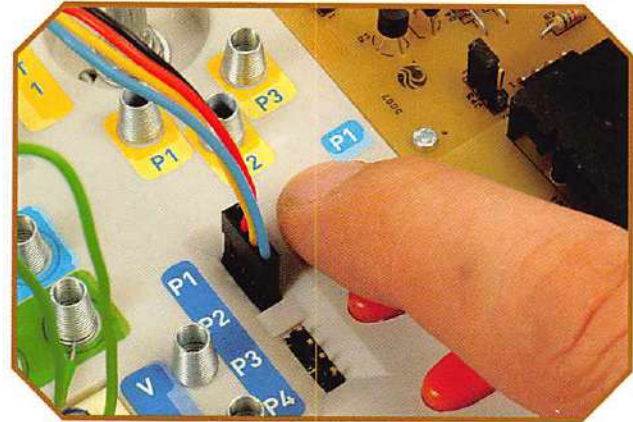
Collegamenti alle molle e all'alimentazione.



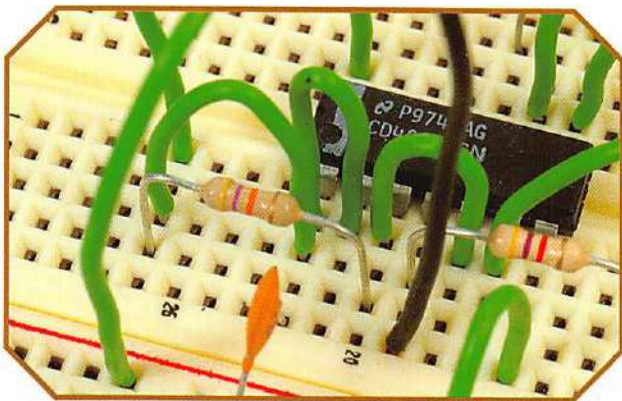
Circuito in funzione.



Premendo P2 si ferma il contatore.



P1 si utilizza per verificare che il conteggio passi per tutti i numeri.



Dettaglio della resistenza R2, sostituita con una da 47 K.

conta molto più lentamente, in modo che sia possibile leggere il display e verificare che il contatore passi per tutti i numeri e li presenti sul display. Se riteniamo che il conteggio sia troppo lento, possiamo sostituire la resistenza R2 con una da 47 K. Nel normale funzionamento P1 non deve essere premuto.

Quando si preme P2 e per tutto il tempo che si mantiene premuto, il contatore si ferma e si potrà leggere il display, la casualità si ottiene, perché è impossibile con la semplice osservazione determinare in quale posizione si trovi il contatore nel momento in cui viene premuto P2.



#### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
D1	Diodo 1N4148
R1	Resistenza 100 K (marrone, nero, giallo)
R2,R4	Resistenza 4K7 (giallo, viola, rosso)
R3	Resistenza 1M (marrone, nero, verde)
C1,C2	Condensatore 22 nF
C3,C4	Condensatore 10 $\mu$ F elettrolitico
LEDA	Diodo rosso 5 mm

Vista del laboratorio con l'esperimento completato.



# Contasecondi elettronico

Questo circuito si utilizza per costruire un clock, in modo che sul display vengano indicati i secondi. Dispone di un pulsante P1 per l'azzeramento e l'inizio del conteggio, e di un altro, P2, per fermare il conteggio. Il potenziometro POT1 è il componente che permette di regolare il circuito per farlo funzionare nel modo desiderato.

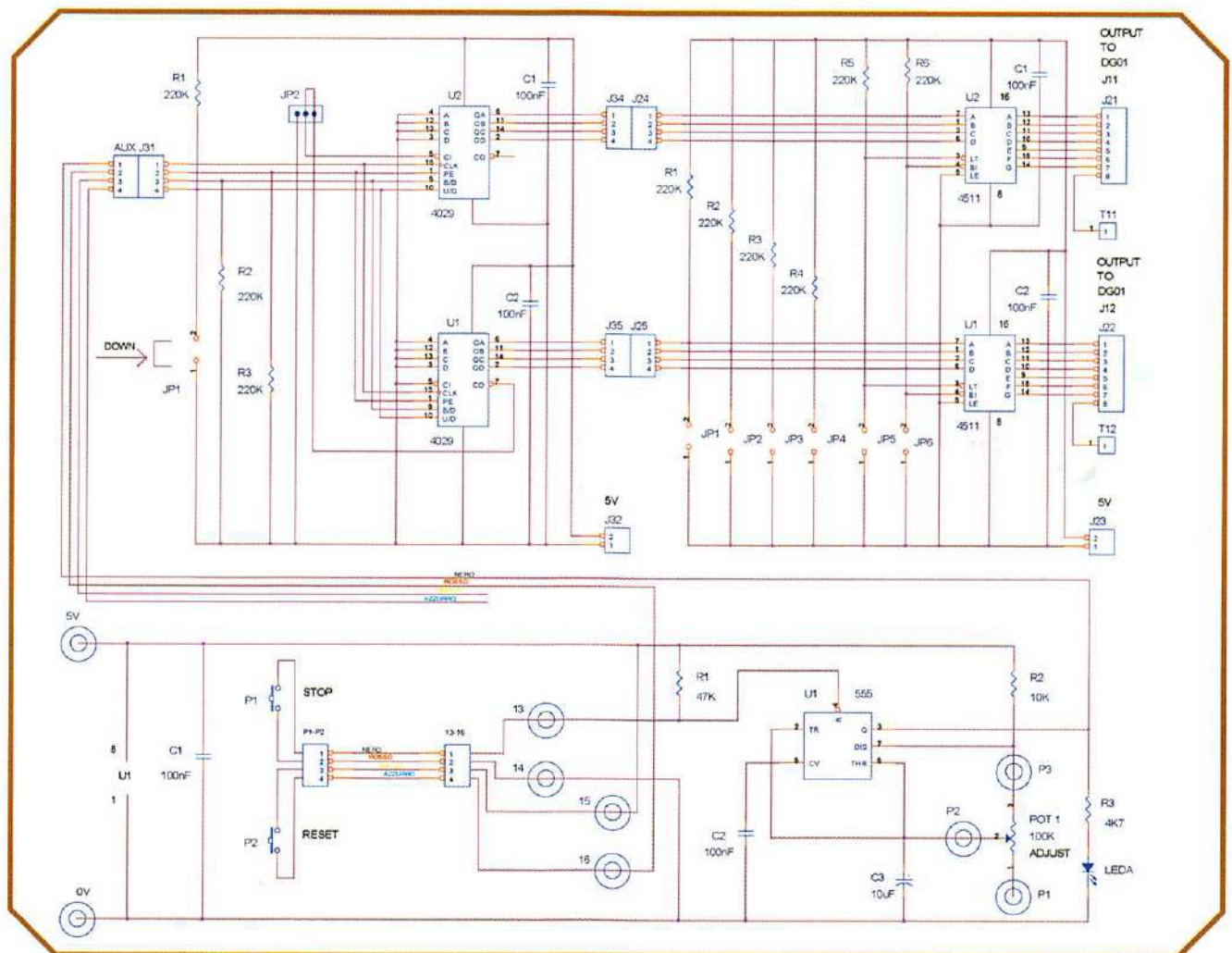
## Il circuito

Lo schema di questo circuito vi risulterà familiare, dato che utilizza il contatore formato dalle schede DG01, DG02 e DG03, montate sul laboratorio. Il circuito di clock è costituito da un circuito integrato 555 configurato come

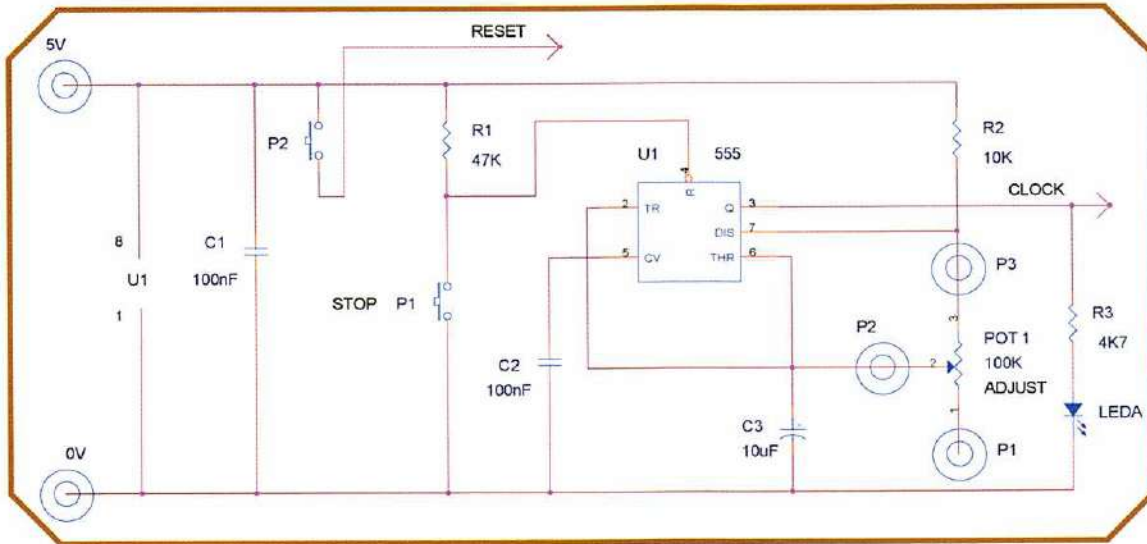
astabile. La sua frequenza di oscillazione si ottiene con la seguente formula approssimata:

$$F = 1,44 / [(R2 + 2RP)C3]$$

Per utilizzare questa formula in modo corretto è necessario esprimere la resistenza in



Schema elettrico del contasecondi.



Schema semplificato del generatore di clock.

Ohm e la capacità in Farad, in modo che il risultato si ottenga in Hertz (Hz). RP rappresenta, nella formula, la resistenza presente tra i terminali P2 e P3 del potenziometro. Questa resistenza dipende dalla posizione del cursore del potenziometro stesso.

Se scegliamo la capacità di 10  $\mu\text{F}$  per il condensatore C3, con R2 da 10 K e la frequenza di 1 Hz, è possibile dedurre che la resistenza necessaria per RP debba essere da 67 K. Utilizzeremo quindi il potenziometro POT1, che vale 100 K, non dobbiamo dimenticare che questo valore è solamente approssimato sia per quanto riguarda la formula che per la tolleranza dei componenti, infatti le resistenze hanno una precisione del 5% e la tolleranza negli elettrolitici può arrivare al 20%.

Il pulsante P1 ferma il funzionamento del-

l'astabile e il contatore ferma il suo conteggio in quanto non riceve più impulsi sull'ingresso del clock, rilasciando P1 il conteggio continua.

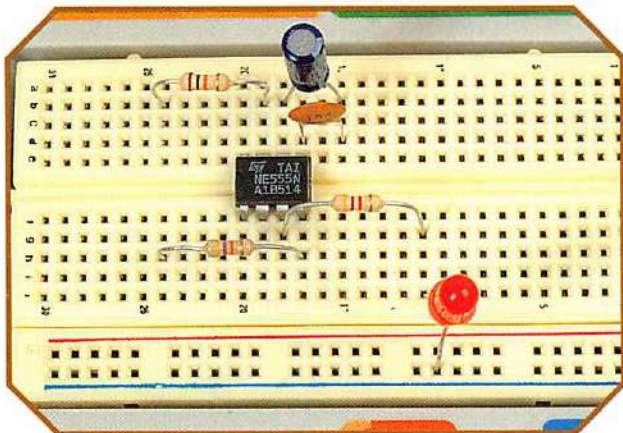
Il pulsante P2 imposta a zero il contatore e quando lo rilasceremo il conteggio inizierà da 00.

Il LED A svolge la funzione di indicatore di funzionamento del clock.

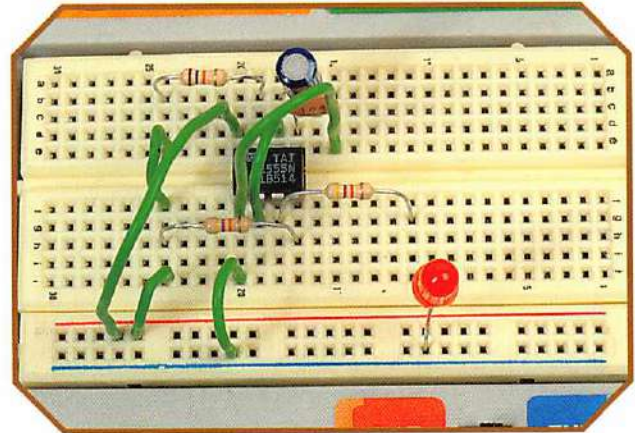
## Montaggio

Il montaggio inizia dall'inserimento del circuito integrato 555 e degli altri componenti sulla scheda Bread Board, tenendo presente l'orientamento del circuito integrato e la polarità del diodo LED A e del condensatore C3.

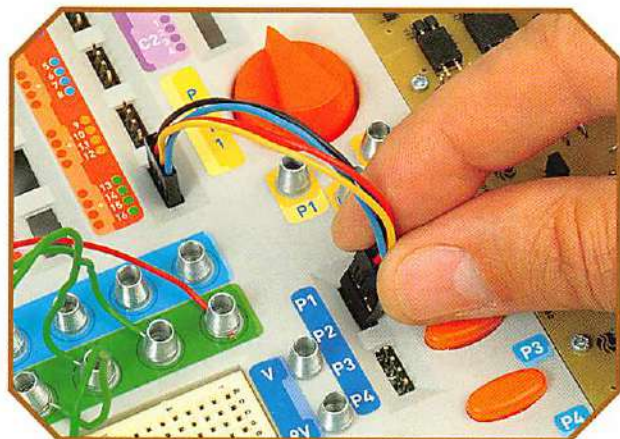
I pulsanti si collegheranno con un cavetto a quattro fili tra il connettore siglato con P1 P2, e il connettore corrispondente alle molle 13-16.



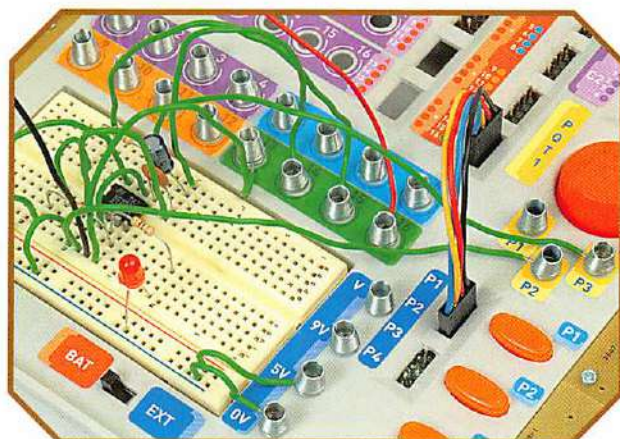
Componenti sulla scheda Bread Board.



Cablaggio interno della scheda Bread Board.



Collegamento dei pulsanti.



Collegamento del potenziometro utilizzato per la regolazione.

Non dobbiamo dimenticare di collegare il potenziometro e l'alimentazione dell'integrato, e come sempre, vi consigliamo di riservare come ultima operazione il collegamento del positivo dell'alimentazione. Il ponticello JP2 della scheda DG03 deve essere inserito in modo che il contatore conti le decine.

Questo circuito si può alimentare indistintamente a 5 oppure a 9 V, tuttavia è necessario verificare che sui ponticelli delle schede DG04 e DG05 sia stata scelta la stessa tensione per le schede del contatore DG01, DG02 e DG03.

## Funzionamento

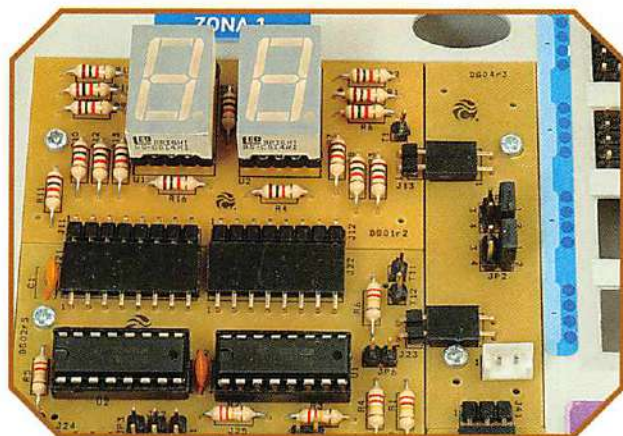
Dopo aver verificato tutto il lavoro si posiziona il comando del potenziometro POT1 approssimativamente alla metà della sua forza e si collega l'alimentazione; in questo momento

il contatore inizierà ad avanzare. Prima di procedere all'alimentazione bisogna verificare che il circuito funzioni.

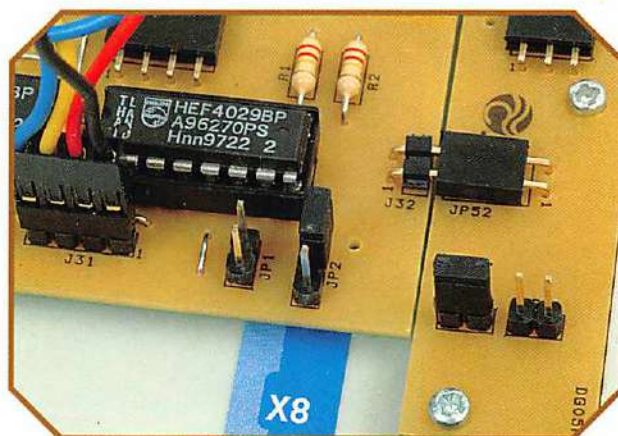
Premendo P1 il conteggio si deve fermare e rilasciandolo deve continuare, premendo P2 il contatore si azzererà e rilasciandolo inizierà a contare da 00. Verificheremo che il conteggio avanzi dallo 00 al 10, questo ci indicherà che il ponticello JP2 sarà stato inserito correttamente.

## Regolazione

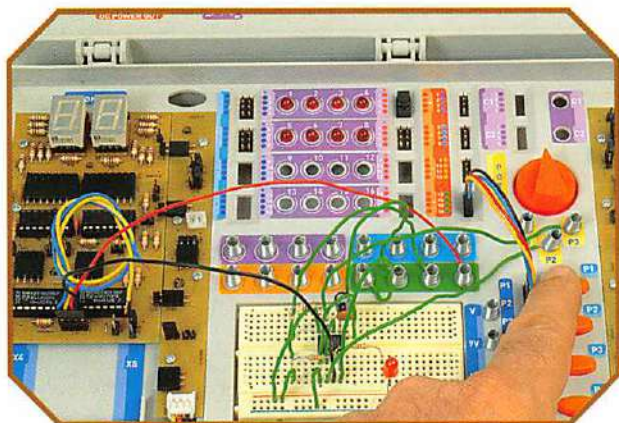
La regolazione va eseguita in modo che la frequenza di oscillazione sia di un secondo. Avremo bisogno di un cronometro o almeno di un orologio con il contasecondi; dato che regolare secondo per secondo è molto difficile, converrà iniziare ad approssimare il valore a occhio, dopodiché misurare un minuto completo con



Dobbiamo fare attenzione al posizionamento dei ponticelli sulla scheda DG04.



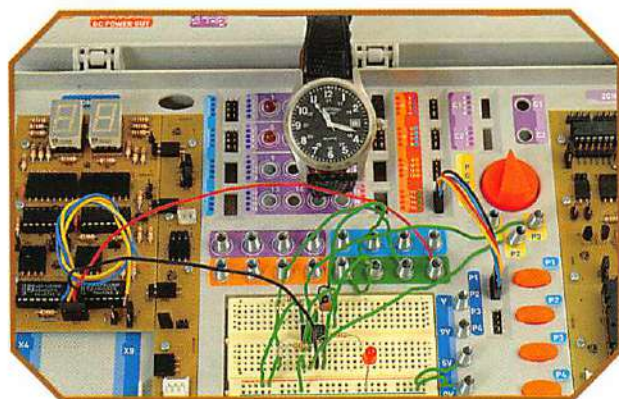
Dettaglio che indica la posizione del ponticello JP2 di DG03 e del ponticello di alimentazione di DG05.



Premendo P1 si ferma il contatore.



P2 si utilizza per impostare a 0 e iniziare il conteggio.



La regolazione si esegue con POT1.

l'orologio utilizzato per la calibrazione. La misura inizierà quando rilasceremo P2 e premeremo poi P1 quando abbiamo raggiunto i 60 secondi; a questo punto verificheremo se il conteggio del display segnerà o meno 60. Ruoteremo un po' il cursore del potenziometro fino a ottenere la regolazione, nel caso in cui la corsa del potenziometro non fosse sufficiente per abbassare la frequenza potremo aumentare il valore della resistenza R2 fino al valore necessario, ad esempio 47 K o 100 K.

Bisogna tener presente che aumentando il valore dei componenti R2, C3 o RP otterremo come risultato un rallentamento del clock.



#### LISTA DEI COMPONENTI

U1	Circuito integrato 555
R1	Resistenza 47 K (giallo, viola, arancio)
R2	Resistenza 10 K (marrone, nero, arancio)
R3	Resistenza 4K7 (giallo, viola, rosso)
C1,C2	Condensatore 100 nF
C3	Condensatore 10 $\mu$ F
LEDA	Diodo LED 5 mm

Vista del laboratorio con l'esperienza completata.





# Finestra temporale

**C**on questo titolo diamo il nome a un circuito che permette il passaggio di un segnale per un certo periodo di tempo, e se questo segnale è alternato, conta quanti impulsi ci sono all'interno di questa "finestra temporale".

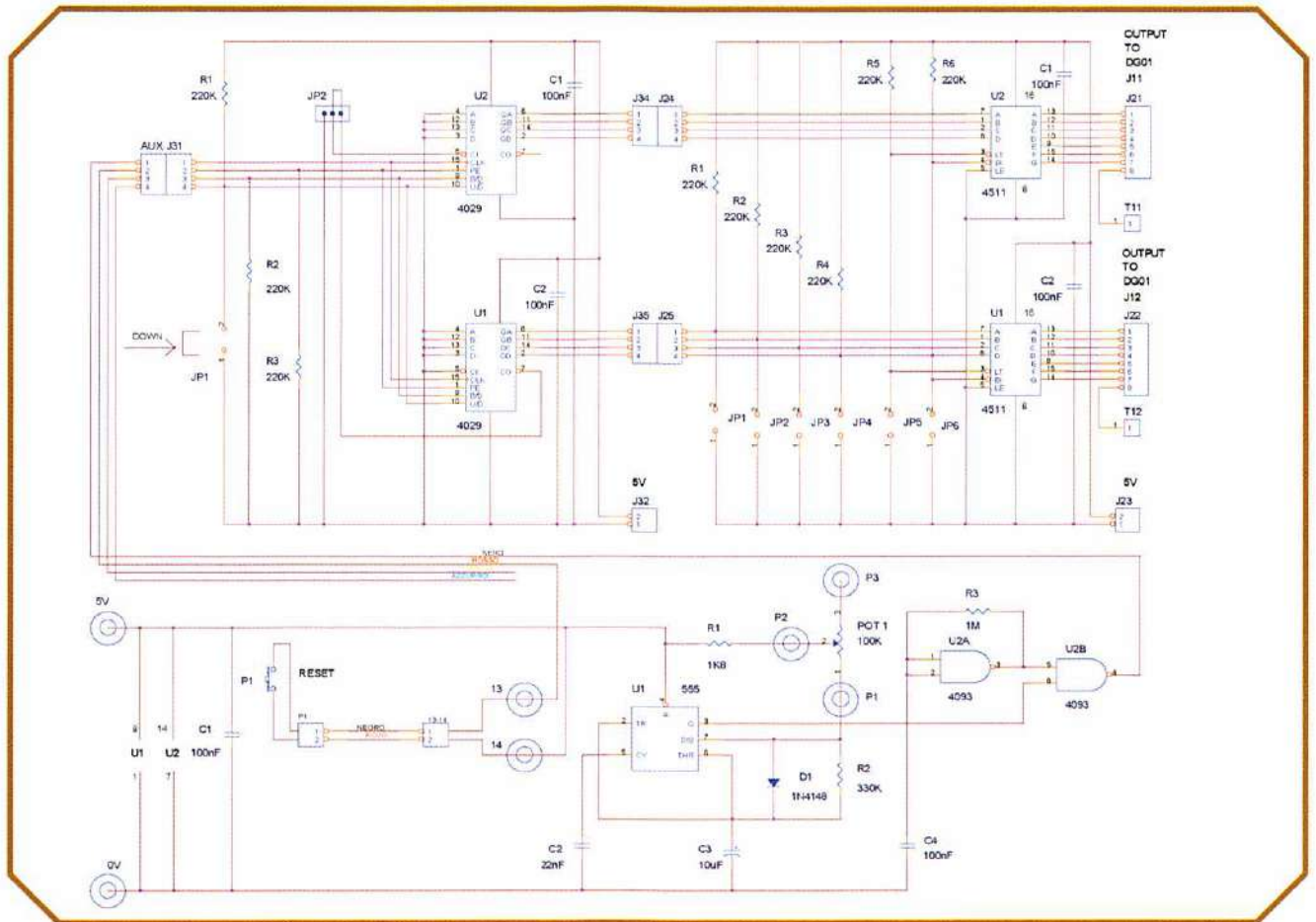
## Il circuito

Osservando lo schema, vediamo rappresentato nella parte superiore il nostro contatore a due cifre. Nella parte inferiore potremo vedere che la porta U2A – resistenza R3 e oscillatore C4 – formano un oscillatore astabile che genera impulsi, i quali devono attraversare la porta U2B per arrivare all'ingresso del clock del contatore, terminale 1 del connettore J31 della scheda DG03. Questa porta lascia passare gli impulsi quando al suo ingres-

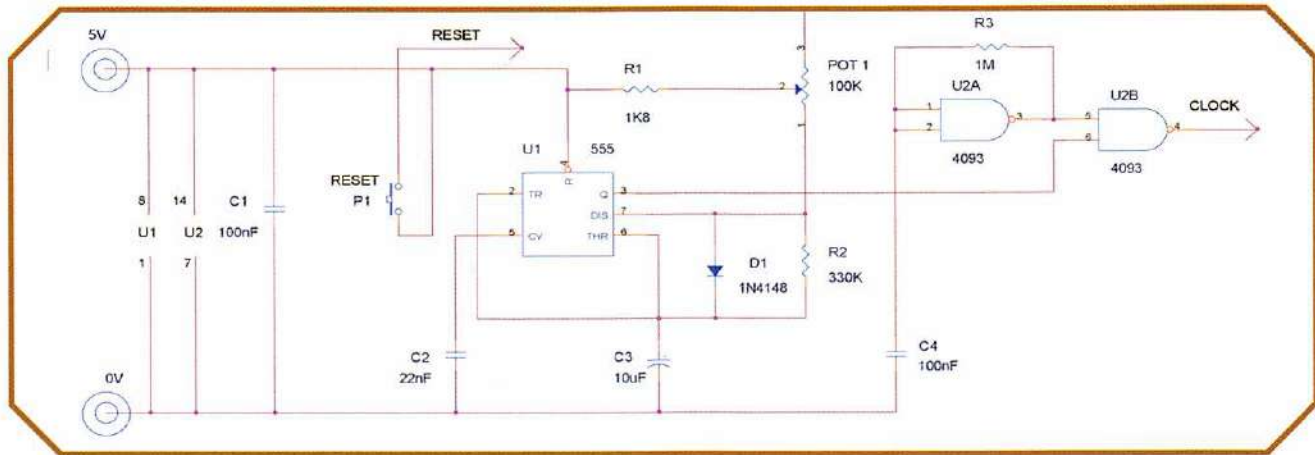
so, terminale 6 di U2, esiste un livello alto, più precisamente il tempo in cui questo ingresso rimane a livello alto è la "finestra temporale".

Questa finestra temporale è controllata da un altro generatore astabile, formato dal circuito integrato U1 e i suoi componenti associati. Il tempo per il quale l'uscita di questo oscillatore astabile è a livello basso è determinato dalla seguente formula:

$$T2 = 0,7 * R2 * C3$$



Schema elettrico del generatore della finestra temporale.



Schema semplificato del circuito.

Il tempo in cui l'uscita del segnale è a livello alto si chiama T1, e viene calcolata così :

$$T1 = 0,7 * (R1+P1) * C3$$

P1 è il valore fra il terminale P1 e P2 del potenziometro e si può regolare tra 0 K e 100 K.

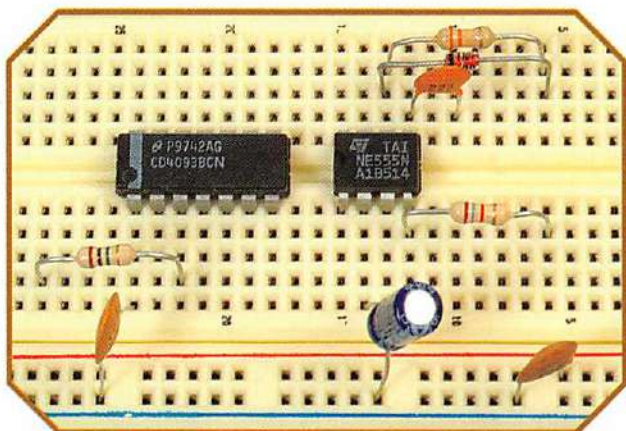
Con il potenziometro quasi a 0, T1 è molto piccolo e passeranno solamente pochissimi impulsi dell'oscillatore verso l'ingresso del contatore, in quanto la finestra rimarrà aperta per poco tempo; T2 è più lungo per poter eseguire con maggior facilità l'esperimento. Sul display sarà possibile vedere quanti impulsi passano, dato che corrispondono alla quantità di avanzamento del contatore.

## Frequenzimetro

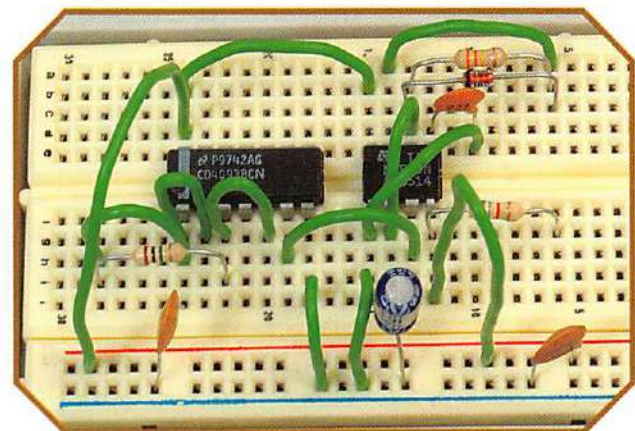
Se aumentiamo il tempo di T1 passeranno più impulsi, perché incrementeremo il tempo della "finestra temporale". Con il pulsante P1 potremo impostare a 0 il contatore, questo ci permetterà di leggere direttamente sul display del contatore la quantità di impulsi che passano.

Se riusciremo a regolare T1 a un valore pari a un secondo, potremo leggere la quantità di impulsi che passano al secondo, cioè la frequenza. Avremo costruito così un semplice frequenzimetro.

Il tempo T2 è utilizzato solamente per separare le finestre temporali; durante questo tempo dobbiamo premere P1 per impostare a 0 il contatore.



Componenti sulla scheda Bread Board.



Cablaggio interno della scheda.



Collegamento del pulsante P1.

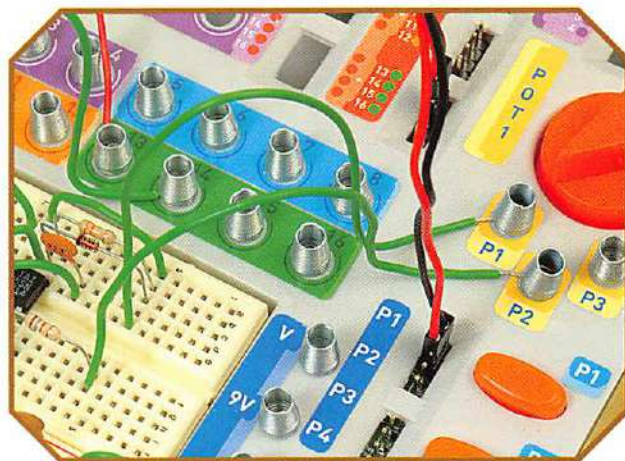
## Montaggio

Il montaggio si esegue come d'abitudine, iniziando con il posizionamento dei componenti sulla scheda Bread Board, facendo attenzione all'orientamento dei circuiti integrati e alla polarità del diodo D1 e del condensatore elettrolitico C3.

Il pulsante P1 necessita solamente di due collegamenti, quindi è sufficiente utilizzare il cavetto a due connettori.

È necessario inoltre collegare i terminali P1 e P2 del potenziometro e le alimentazioni dei circuiti integrati. Il ponticello JP2 della scheda DG03 deve essere inserito per fare in modo che il contatore conti le decine.

Questo circuito si può alimentare indifferentemente a 5 o 9 V, controllando che le sche-

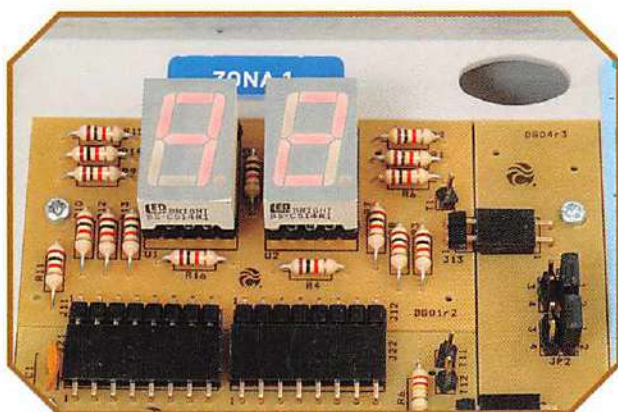


Bisogna collegare i terminali P1 e P2 di POT1.

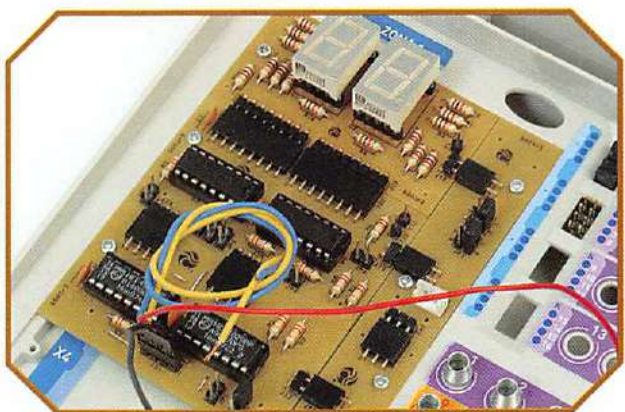
de del contatore siano anch'esse alimentate alla stessa tensione.

## Funzionamento

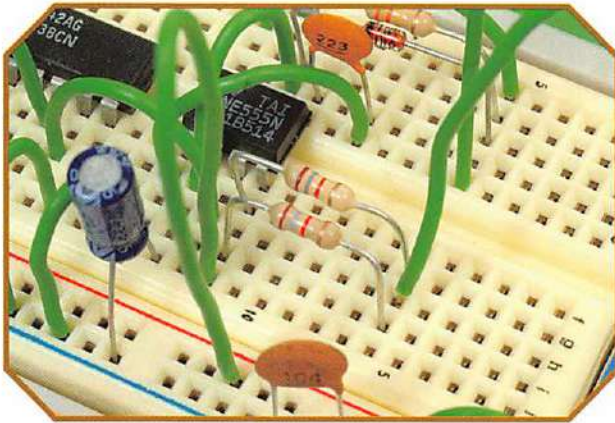
Dopo aver verificato tutto il lavoro si posiziona la manopola del potenziometro POT1 all'inizio della sua corsa e si collega l'alimentazione. In questa situazione il contatore deve incrementare il suo conteggio all'incirca ogni 2 secondi, in altre parole potrebbe avanzare di 1 in 1, di 2 in 2, di 3 in 3, di 4 in 4, ecc. Nel caso in cui il contatore non avanzasse, provate a girare la manopola del potenziometro, in quanto la finestra temporale potrebbe essere troppo piccola comparata con la frequenza del generatore, in questo caso non passerebbe alcun impulso e il contatore non si muoverebbe.



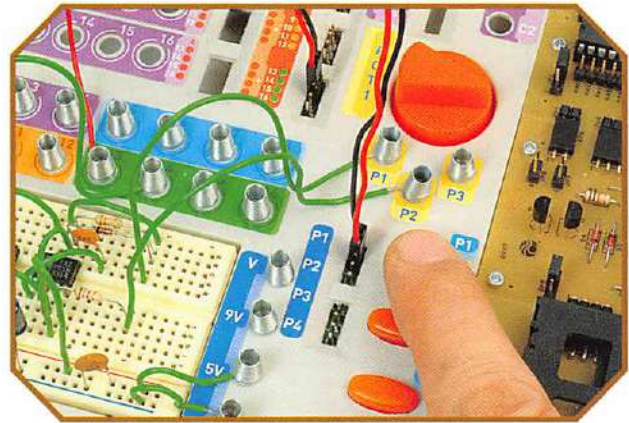
Il display avanza a ogni impulso ricevuto.



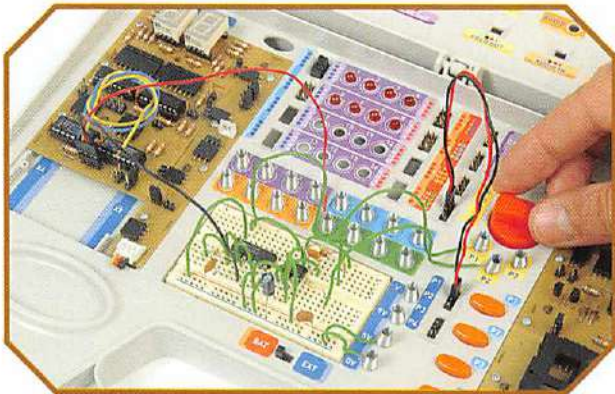
Il contatore può avanzare di 2 in 2, di 3 in 3, di 4 in 4 di 5 in 5, ecc.



R1 si può diminuire aggiungendo un'altra resistenza di uguale valore in parallelo.



P1 imposta a 0 il contatore.

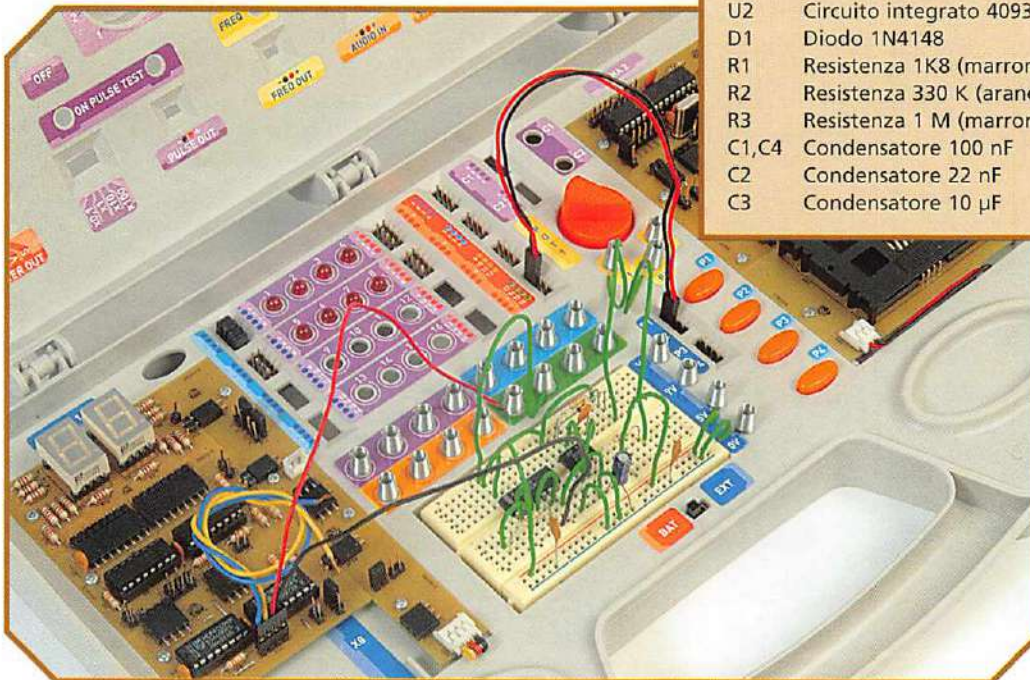


Regolazione del tempo "finestra".

Se premiamo P1 per impostare a 0 il display, possiamo vedere che il contatore incrementerà il suo conteggio per un determinato periodo di tempo, ruotando ulteriormente il comando del potenziometro aumenteremo il tempo a disposizione del contatore per il conteggio. Se riusciamo a regolare il tempo di attività del contatore a 1 secondo, potremo conoscere la frequenza del segnale generato dalla porta U2A.

**LISTA DEI COMPONENTI**

- U1 Circuito integrato 555
- U2 Circuito integrato 4093
- D1 Diodo 1N4148
- R1 Resistenza 1K8 (marrone, grigio, rosso)
- R2 Resistenza 330 K (arancio, arancio, giallo)
- R3 Resistenza 1 M (marrone, nero, verde)
- C1,C4 Condensatore 100 nF
- C2 Condensatore 22 nF
- C3 Condensatore 10 µF

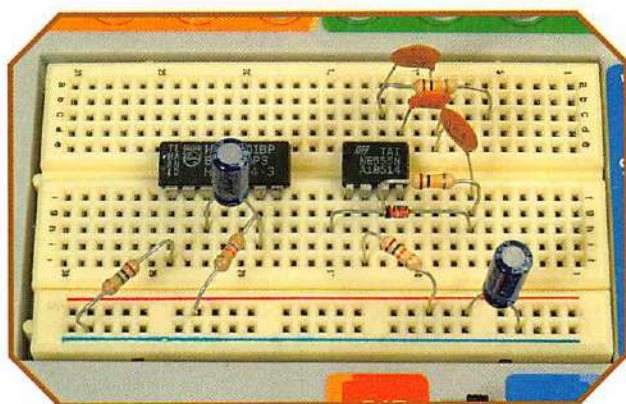


Vista del laboratorio con l'esperimento completato.

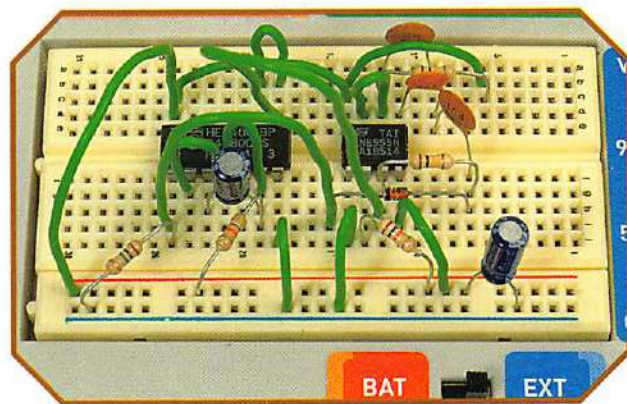


# Contatore bidirezionale

**I**n questo circuito si controlla un contatore con tre pulsanti: uno produce il conteggio, l'altro provoca la diminuzione e il terzo lo azzerera completamente.



Componenti sulla scheda Bread Board.



Cablaggio interno della scheda.

## Scopo del circuito

Vogliamo un circuito che funzioni nel seguente modo: ogni volta che si preme un pulsante il contatore avanzerà di una unità mentre se ne azioniamo un altro retrocederà di una unità, tutto questo in modo automatico e senza bisogno di dover cambiare i ponticelli o i collegamenti; inoltre un terzo pulsante azzererà il conteggio.

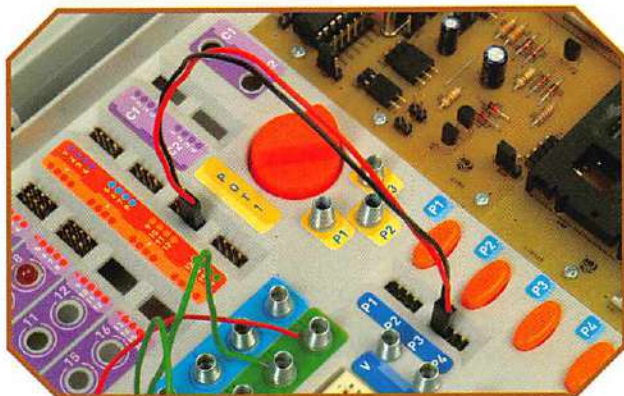
Inizieremo il progetto seguendo un metodo leggermente diverso da quanto fatto finora. Il pulsante P1 azionerà un monostabile la cui uscita applicherà un unico impulso all'in-

gresso di clock del contatore. Per il pulsante P3 sarà necessario utilizzare un circuito addizionale per impostare a livello basso il terminale 2 del connettore J31 della scheda DG03, prima che l'impulso di clock arrivi all'ingresso di clock del contatore; in questo modo il contatore conterà a ritroso.

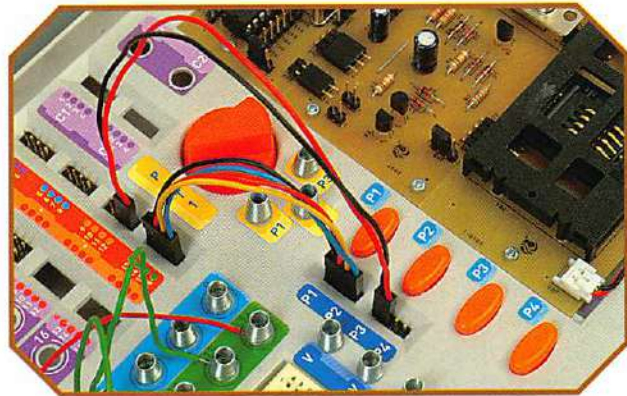
Vediamo ora, con l'aiuto dello schema, come si realizza il circuito.

## Schema

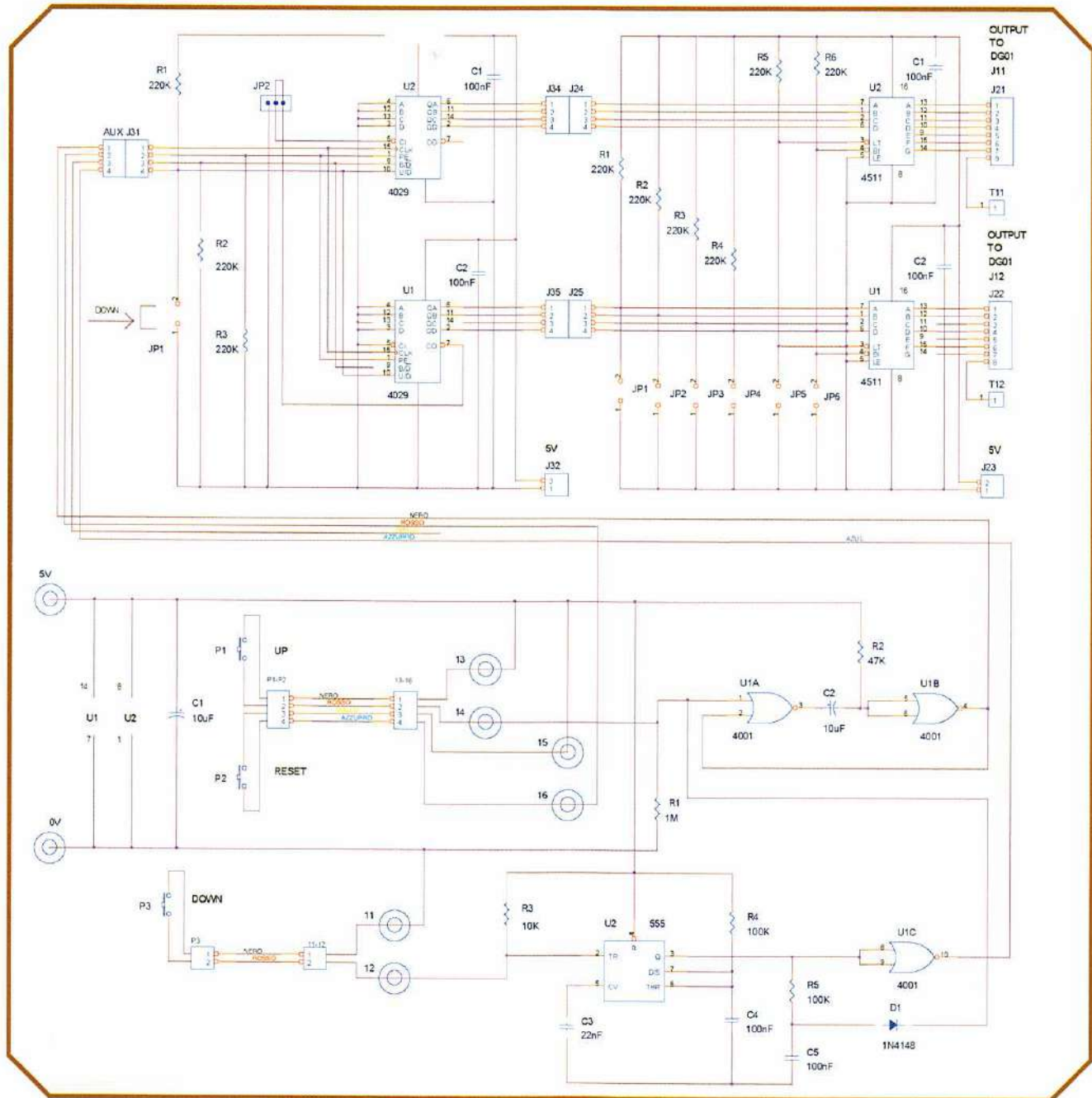
Per descrivere lo schema inizieremo dai pulsanti e ne seguiremo il collegamento. Il pul-



Collegamenti del pulsante P3.



Collegamenti dei pulsanti P1 e P2.



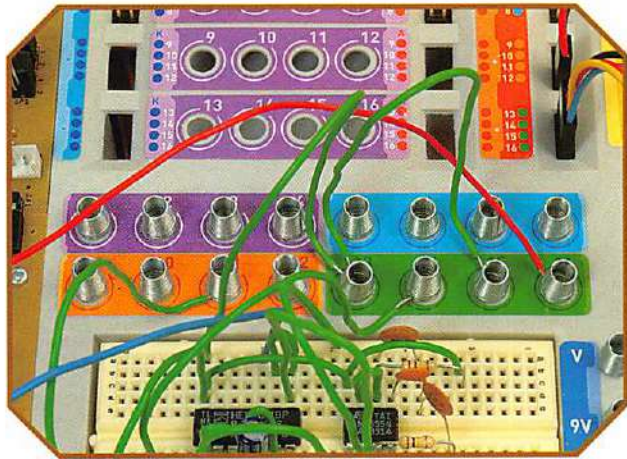
Schema elettrico del contatore.

sante P1 attiva il monostabile formato dalle porte U1A e U1B del circuito integrato 4001. Ogni volta che si preme si applica un impulso al terminale 1 di J31 e il contatore avanza di una unità.

Ogni volta che si aziona P2 si imposta a zero il contatore.

Analizziamo ora il funzionamento del pul-

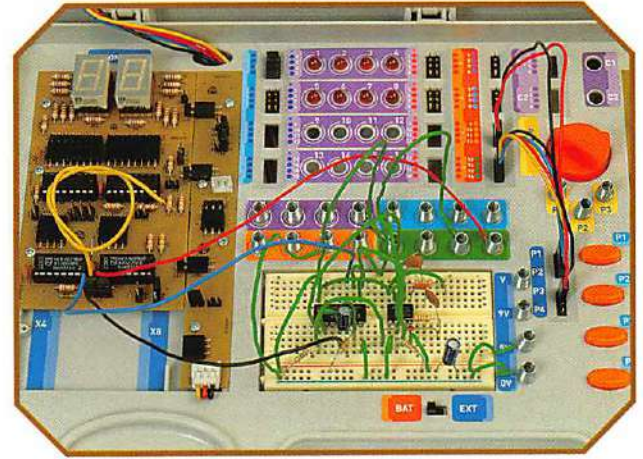
sante P3. Quando si aziona questo pulsante si attiva il monostabile formato dal circuito integrato U2, che è un 555 configurato come monostabile. L'uscita di questo circuito, che in condizioni di riposo è a livello basso, passa a livello alto e tramite una porta invertente arriva al terminale 4 di J31; quando questo terminale è a livello basso, se il contatore ri-



*I collegamenti dei pulsanti sono disponibili sulle molle dalla 11 alla 16.*

ceve un impulso di clock il conteggio si riduce. Ogni impulso del monostabile che inverte il senso del conteggio, terminale 3 del 555, si trasmette all'ingresso del monostabile che applica impulsi di clock, cioè arriva al terminale 1 di U1A, ma con il ritardo provocato dalla rete di ritardo RC – formata dal condensatore C4 e dalla resistenza R4 – in modo che l'impulso che inverte il senso del conteggio arrivi prima al contatore dell'impulso di clock.

Dopo un determinato periodo di tempo il contatore torna allo stato di riposo: se si preme P1 conta in avanti, se si preme P3 conta indietro. La resistenza R1 scarica il condensatore C5 dopo aver prodotto l'attivazione, e il diodo D1 evita che gli impulsi di attivazione



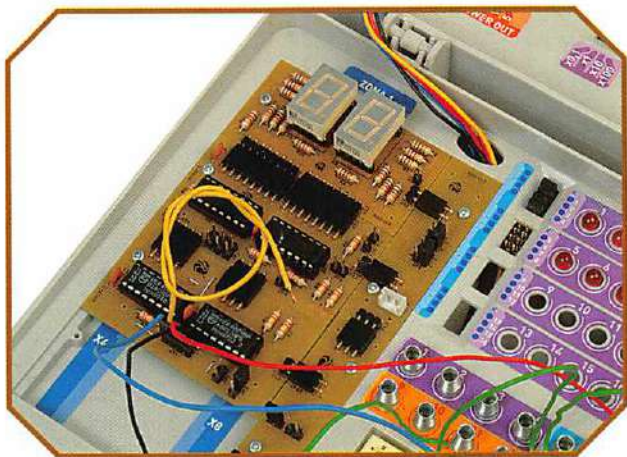
*Circuito pronto per la prova.*

generati da P1 possano arrivare all'ingresso della porta U1C.

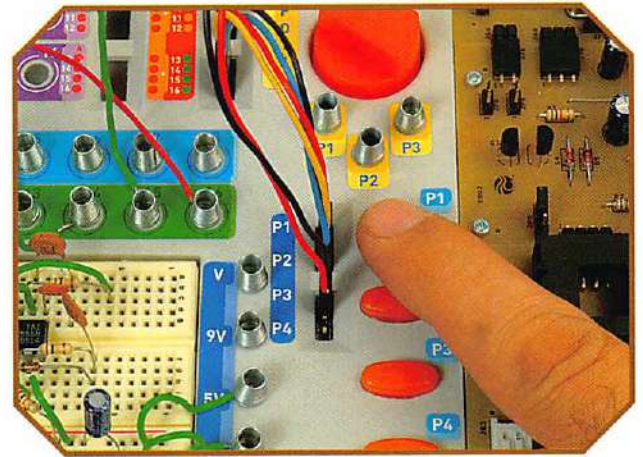
## Montaggio

Il montaggio si realizza come d'abitudine, iniziando con l'inserimento dei componenti sulla scheda Bread Board, tenendo presente l'orientamento dei circuiti integrati e la polarità del diodo D1 e del condensatore elettrolitico C2.

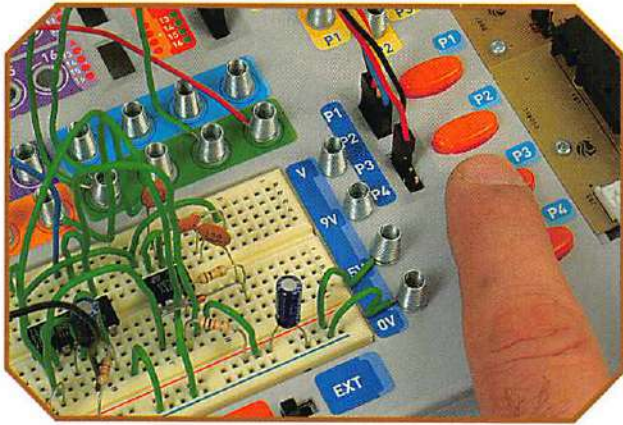
Il pulsante P3 (DOWN) necessita solamente di due connessioni e si collega con il cavetto a due fili. I pulsanti P1 (UP) e P2 (RESET) si collegano mediante un cavetto a quattro fili. Il ponticello JP2 della scheda DG03 deve essere montato in modo che il contatore conti le de-



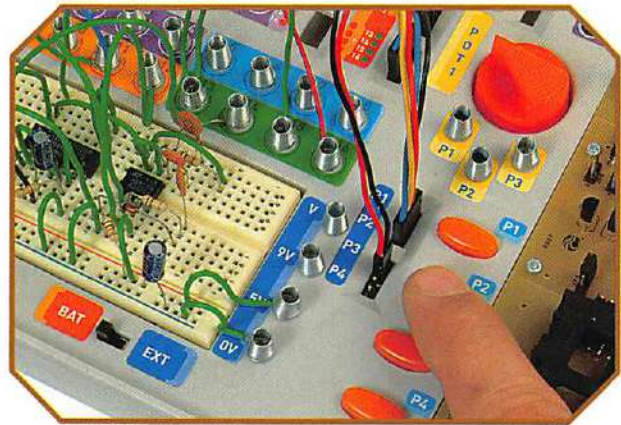
*Contatore predisposto per l'esperimento.*



*Ogni volta che si preme P1 il contatore avanza.*



Premendo P3 il conteggio diminuisce di una unità.



Il pulsante P2 si utilizza per azzerare il conteggio.

cine. I collegamenti al terminale J31 del contatore si eseguono con un cavetto terminato da un lato su un connettore, e dall'altro lato con i fili liberi spelati in punta.

Questo circuito si può alimentare indistintamente a 5 oppure a 9 V, verificando che le schede del contatore siano anch'esse alimentate alla tensione scelta.

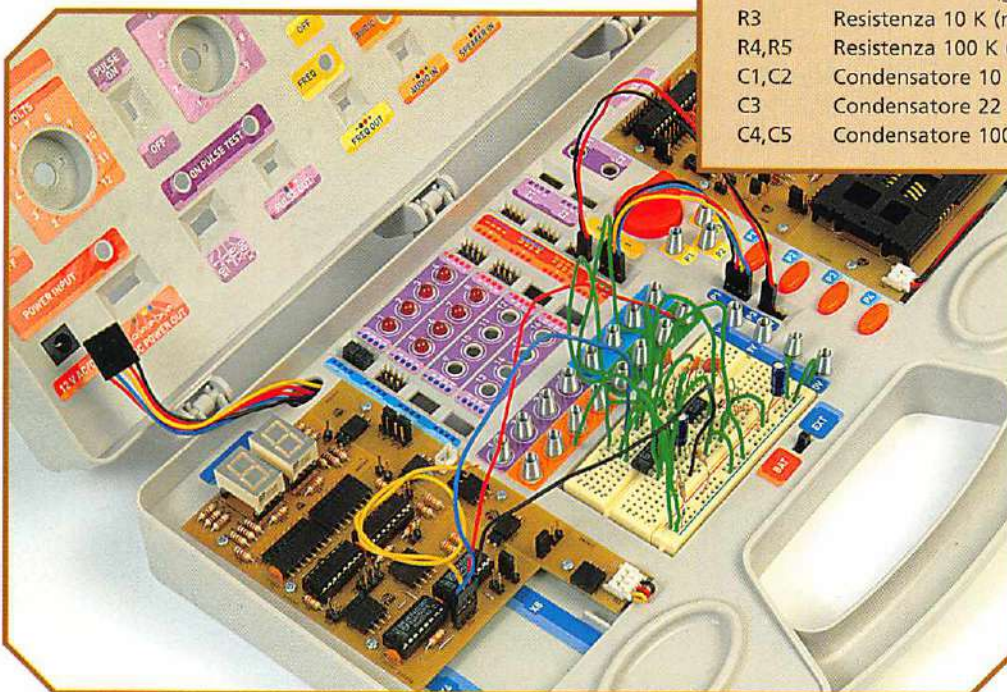
## Funzionamento

Dopo aver verificato il lavoro svolto, si collega l'alimentazione e il circuito risulta pronto per

il funzionamento. Se premiamo P1 il contatore avanza di una unità. Premendo varie volte il contatore avanzerà di diverse unità. Se successivamente premeremo P3 il contatore avanzerà in senso inverso, e se premeremo P2 il contatore si resetterà.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4001
U2	Circuito integrato 555
D1	Diodo 1N4148
R1	Resistenza 1 M (marrone, nero, verde)
R2	Resistenza 47 K (giallo, viola, arancio)
R3	Resistenza 10 K (marrone, nero, arancio)
R4,R5	Resistenza 100 K (marrone, nero, giallo)
C1,C2	Condensatore 10 $\mu$ F
C3	Condensatore 22 nF
C4,C5	Condensatore 100 nF



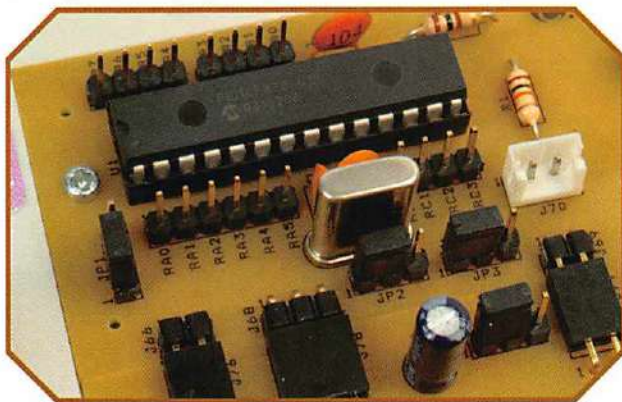
Lista del laboratorio con l'esperimento completato.



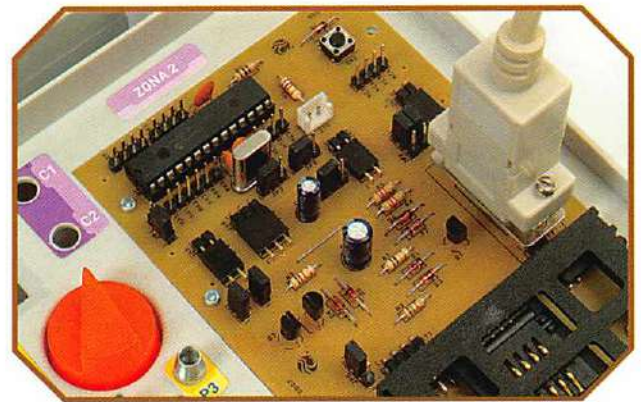


## Esercizi 7, 8 e 9: la pratica

**N**egli esercizi di ripasso (microcontroller 91 e 92) abbiamo spiegato i programmi per lavorare con i temporizzatori TMR0 e TMR1, abbiamo progettato, compilato e anche simulato il codice, però ci manca ancora la cosa più importante: verificare il suo funzionamento nella pratica e che effettivamente esegua ciò che ci attendiamo. Dobbiamo quindi scrivere i programmi sul microcontroller ed eseguire i relativi montaggi.



Dobbiamo configurare i ponticelli della scheda per poter eseguire la scrittura.



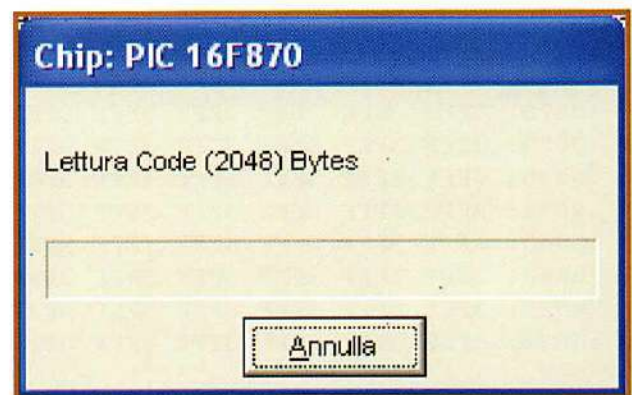
Dopo aver collegato il Laboratorio al PC saremo pronti per scrivere.

### Configurazione hardware per IC-Prog

Per scrivere i programmi sul microcontroller utilizziamo il software IC-Prog, è necessario però che il laboratorio sia correttamente configurato in modo che il trasferimento si realizzi correttamente. Il microcontroller deve essere inserito sullo zoccolo nella posizione corretta e i connettori della scheda DG06: JP1, JP2 e JP3, devono avere i ponticelli nelle posizioni 1 e 2. Ricordate che per qualsiasi altra applicazione del microcontroller i ponticelli di questi connettori verranno posizionati sui terminali 2 e 3. La scheda avrà, in questo modo, la configurazione che possiamo vedere nella figura. I ponticelli JP8 e JP9 devono essere anch'essi collegati per scrivere il PIC e, in ultimo, dobbiamo anche collegare il cavo di scrittura tra il laboratorio e il PC.

### IC-Prog

Inizieremo facendo partire IC-Prog e verificando di aver selezionato il PIC16F870 e che la



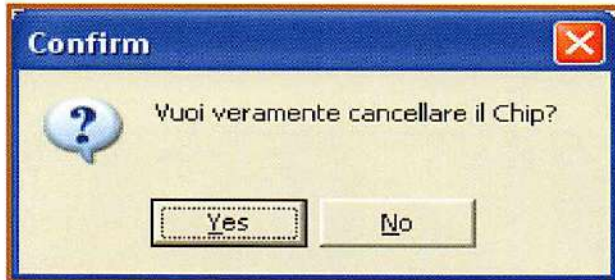
Finestra che indica lo stato del processo di lettura.



Il programma ci avvisa che il PIC è stato cancellato correttamente.



scrittura venga eseguita utilizzando la porta desiderata. Se osserviamo la parte inferiore della finestra del programma potremo vedere il programmatore e il dispositivo selezionato, nel nostro caso: "JDM Programmer su Com1"

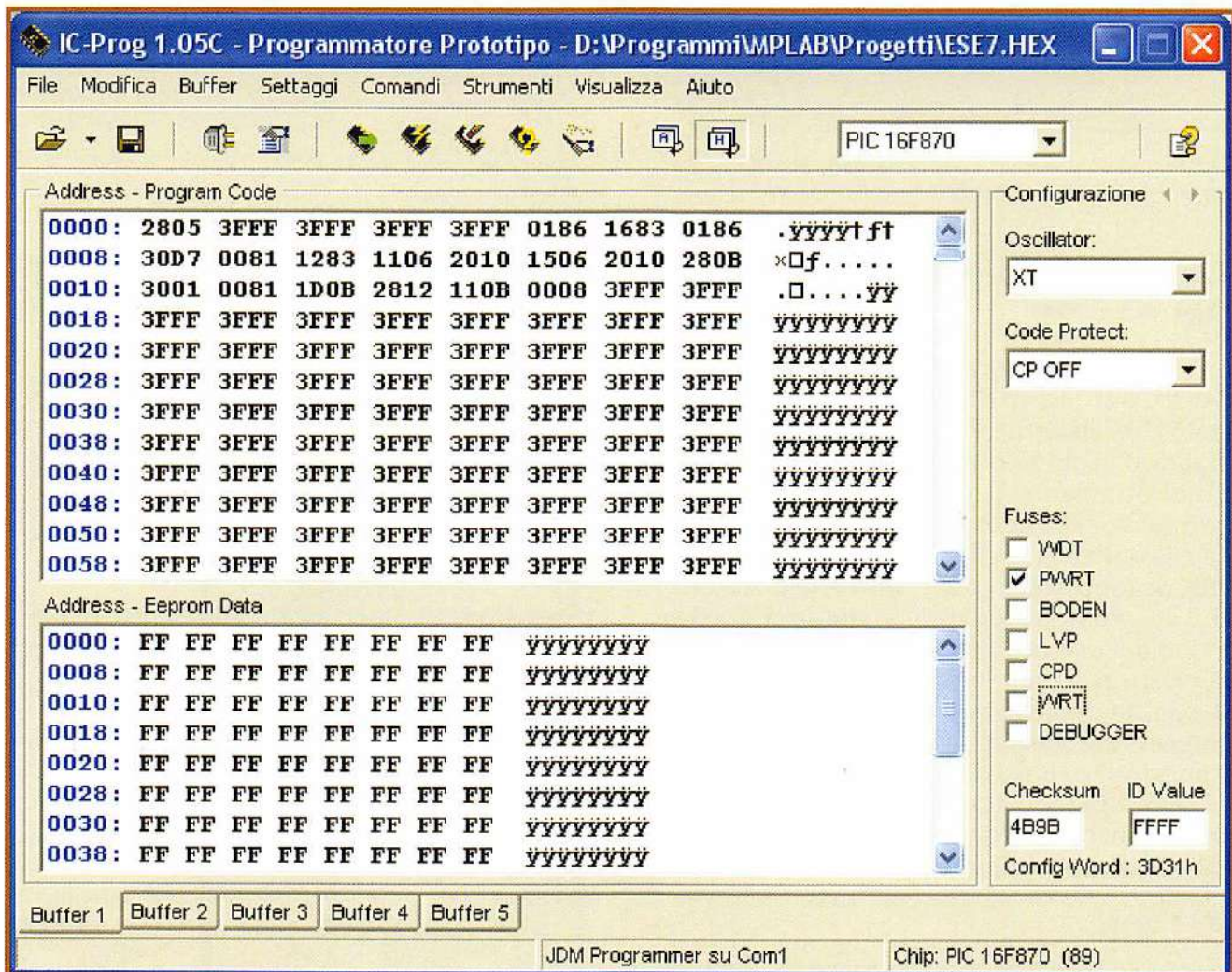


Quando selezioniamo cancella e programma dobbiamo confermare la scelta.

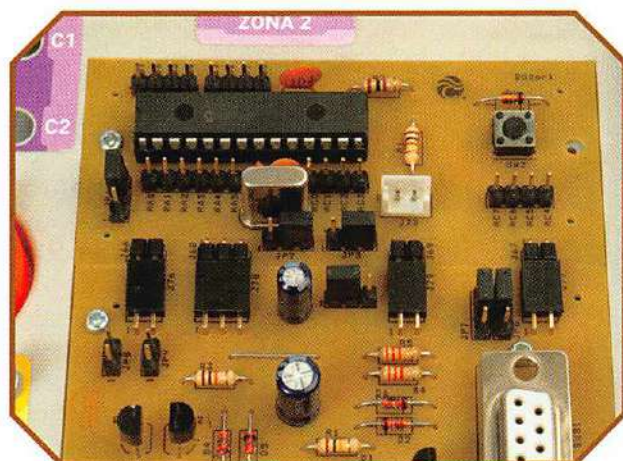
e "Dispositivo: PIC 16F870 (89)". Se non compaiono questi valori dovremo configurare il programma adeguatamente.

La prima operazione che dovremo eseguire sul PIC sarà leggere il programma che esso contiene. A questo scopo selezioneremo l'opzione Leggi tutto dal menù Comandi, oppure premeremo F8 o selezioneremo l'icona corrispondente sulla barra degli strumenti. Se il PIC contiene un programma memorizzato cambieranno i valori della finestra "Program code". Il programma letto potrà essere salvato sul nostro PC se lo riterremo opportuno.

L'operazione successiva sarà cancellare il dispositivo, passaggio consigliabile prima di scrivere il programma. Selezioneremo a questo scopo Cancella tutto dal menù o con l'ico-



Aspetto di IC-Prog debitamente configurato e caricato con l'esercizio "ese7.hex".

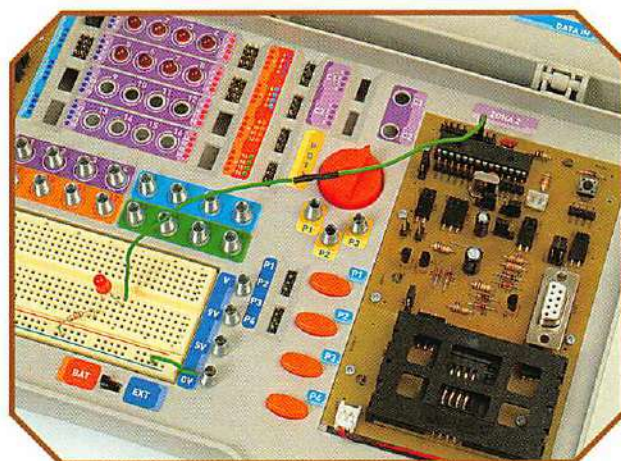


Configurazione dei jumper per il funzionamento normale o di lavoro.

na corrispondente sulla barra degli strumenti. Essendo questo un processo critico, apparirà una richiesta di conferma a cui risponderemo Yes, e in questo modo si cancellerà il dispositivo. Al termine di questo processo apparirà una finestra come quella riportata in precedenza che ci confermerà il successo dell'operazione. Per verificare il processo leggeremo nuovamente il dispositivo e controlleremo che in tutti gli indirizzi di memoria della finestra del codice di programma appaia il valore 3FFF. In alcuni casi è necessario configurare l'oscillatore e i bit della parola di configurazione, prima di procedere alla cancellazione, anche se questo non è molto comune. La corretta impostazione della parola di configurazione è invece realmente necessaria per realizzare la scrittura di un programma sul microcontroller.

## Scrittura

Dobbiamo selezionare il programma che vogliamo scrivere sul PIC. Apriamo il file desiderato (.hex) e verifichiamo come cambia la finestra che contiene il codice del programma. Prima di scrivere configuriamo l'oscillatore selezionando XT e la parola di configurazione selezionando i bit WDT e PWRT per l'esercizio "ese8.hex". Ricordate che dovremo mantenere l'opzione di protezione del codice su CP OFF. Selezionando Programma tutto caricheremo il programma selezionato sul PIC. Dato che anche questo è un processo critico ci verrà presentata una richiesta di conferma del pro-



Montiamo il circuito comune ai due esercizi del TMR0.

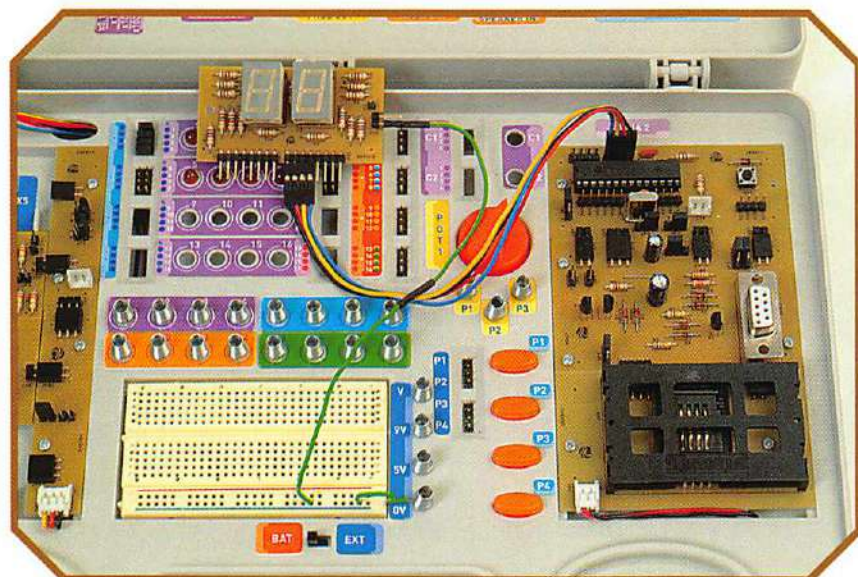
cesso a cui risponderemo Yes. Anche se otterremo un messaggio di verifica corretta, è consigliabile leggere nuovamente il dispositivo per verificare che il processo sia stato eseguito con successo.

Ora abbiamo a disposizione il microcontroller che contiene il programma desiderato. Per caricare qualsiasi altro programma dovremo ripetere i passi visti finora.

## Montaggio dell'esercitazione con il TMR0

I due esercizi predisposti per lavorare con il TMR0 hanno lo stesso montaggio. La prima cosa da tener presente e che per lavorare con il Laboratorio in modo diverso dal trasferimento dei dati con il computer, dobbiamo cambiare la configurazione dei ponticelli. Ora JP1, JP2 e JP3 dovranno avere i ponticelli sui terminali 2 e 3, toglieremo i ponticelli dai JP8 e JP9 e collegheremo il cavo di scrittura dal laboratorio e dal PC. Nelle immagini della figura possiamo osservare come deve essere configurato il laboratorio per lavorare con il microcontroller in modo funzionamento o di lavoro.

Uniremo mediante un filo il terminale RB2 con l'anodo di uno dei due diodi della matrice e collegheremo i ponticelli sui catodi dello stesso. Dopo aver fatto questo potremo fornire alimentazione e verificare se abbiamo risolto correttamente l'esercizio, controllando come lampeggia il diodo alla frequenza programmata.



Montaggio per l'esercitazione con il TMR1 "ese9".

due cavetti di collegamento. Questa volta invece di collegarci alla matrice dei diodi ci collegheremo direttamente a uno di due display, J11 o J12. Nel caso in cui i cavetti non arrivassero al connettore, potremo staccare la scheda e avvicinarla facendo attenzione a non entrare in contatto con qualche elemento metallico che possa provocare un corto circuito. Dopo esserci assicurati che i ponticelli siano nella configurazione corretta

## Esercizio 8

Se carichiamo sul PIC l'altro esercizio del TMR0, ovvero ese8, potremo verificare come anch'esso funzioni correttamente. Ricordate che per caricare il programma sul microcontroller è necessario configurare nuovamente l'hardware del laboratorio, aprire il software IC-Prog e caricare il programma con i passaggi che esso richiede. In questo caso il lampeggio sarà più lento e, la frequenza minore così come volevamo che fosse.

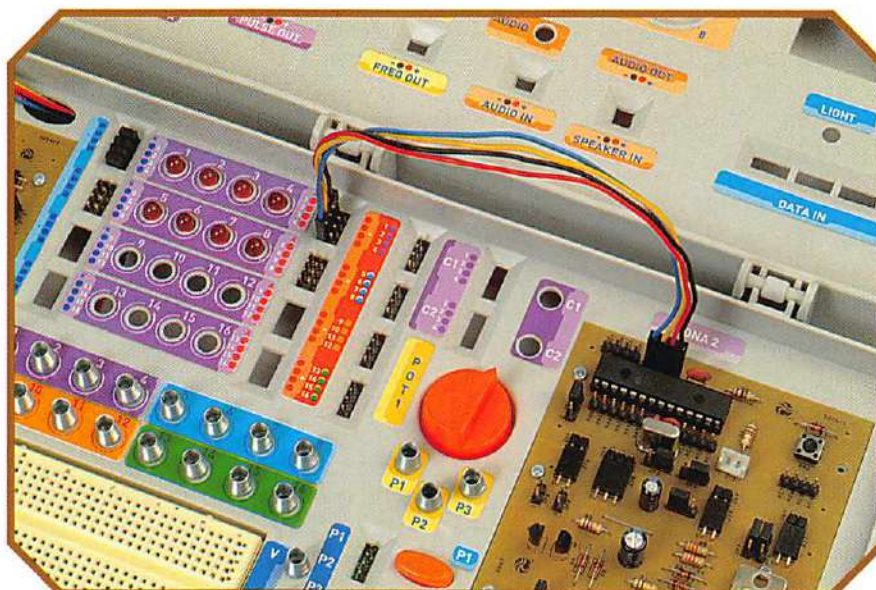
## Montaggio dell'esercitazione con il TMR1

Per verificare la corretta soluzione di questa esercitazione dobbiamo eseguire il relativo montaggio, dopo aver correttamente caricato il programma "ese9" sul PIC. In questo caso utilizzeremo come uscita tutta la porta B, quindi dovremo utilizzare

per il modo funzionamento o modo lavoro (JP1, JP2 e JP3 con i ponticelli sui terminali 2 e 3, e JP8 e JP9 tolti), potremo fornire alimentazione e osservare il risultato.

Vediamo come i LED del display (o i LED della matrice) si accendono in modo sequenziale, rispondendo all'istruzione di incrementare di 1 il valore di uscita, che abbiamo utilizzato nel codice del programma.

Potete provare questo stesso esercizio collegando la porta B alla matrice dei diodi o provare a modificarlo utilizzando, ad esempio, le istruzioni di rotazione.



*Nel laboratorio è più semplice lavorare con la matrice di LED che con il display.*



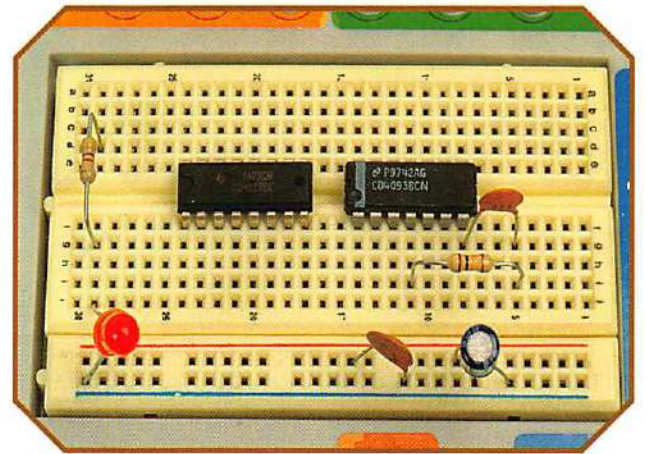
# Contatore di rumore

**C**on questo esperimento vogliamo verificare come un circuito possa essere disturbato dal rumore radioelettrico ambientale. Il rumore captato genera impulsi che fanno funzionare il contatore in modo praticamente incontrollato. Il circuito contiene un diodo LED in più, il LEDA, per evidenziare meglio l'effetto.

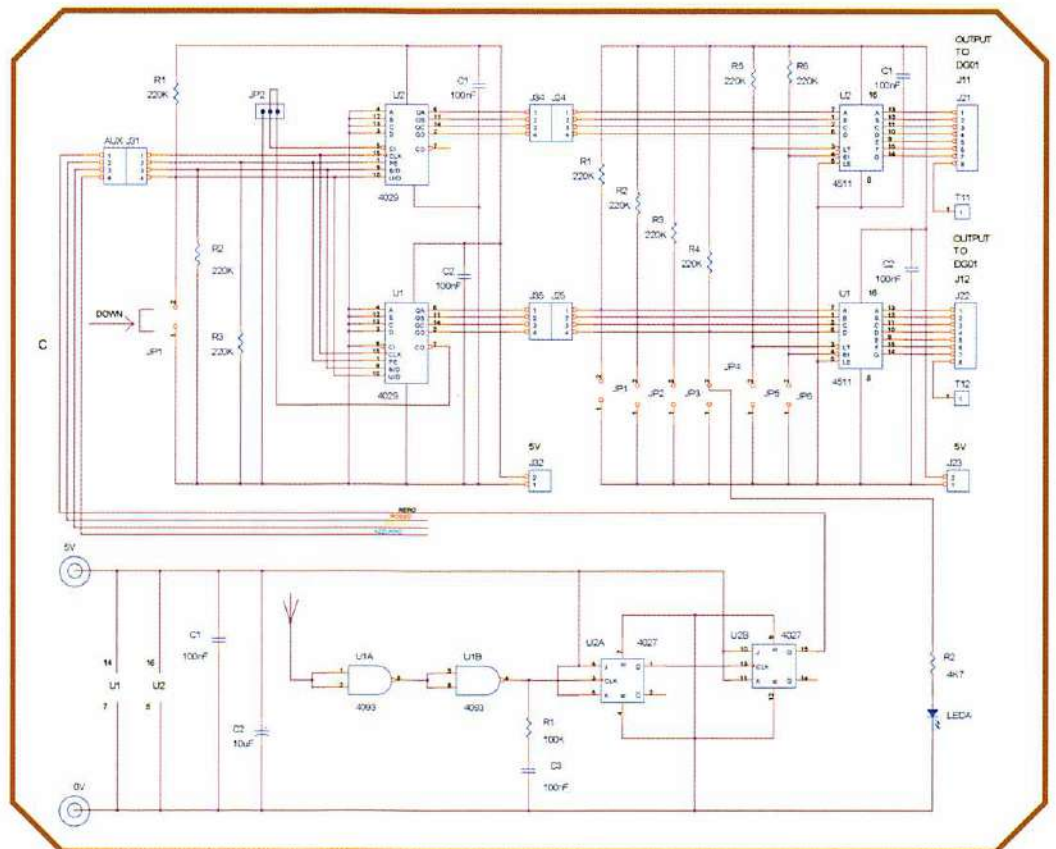
## Il circuito

Nello schema possiamo vedere il contatore a due digit del laboratorio, fatta eccezione per la parte che corrisponde al display, che non è stata rappresentata per non dover diminuire ulteriormente la dimensione dei simboli utilizzati, facilitando così la lettura e lo studio dello schema.

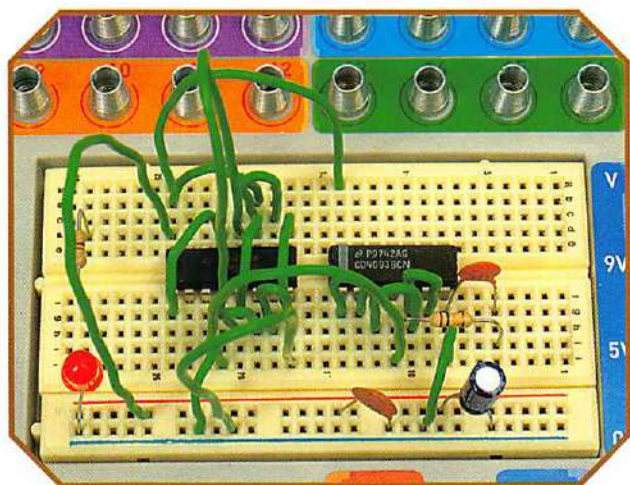
Nella parte inferiore dello schema possiamo vedere il simbolo di un'antenna che si collega alla porta U1A dell'integrato U1. Questa antenna è un pezzo di filo la cui lunghezza non è molto importante per l'esperimento, va comunque sottolineato che lunghezze diverse per l'antenna possono captare disturbi di-



Componenti sulla scheda Bread Board.



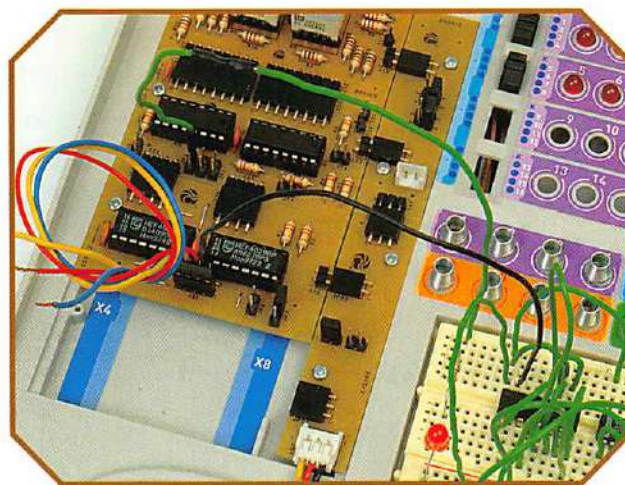
Schema elettrico del rilevatore di rumore.



*Cablaggio interno della scheda Bread Board.*

versi. Uno dei disturbi che si può captare con una certa facilità è quello prodotto dai 50 o 60 Hz della rete di distribuzione dell'energia elettrica, ma bisogna fare attenzione perché devono essere captati tramite l'aria, non bisogna assolutamente realizzare collegamenti alla rete elettrica. La rete formata dalla resistenza R1, collegata in serie al condensatore C3, diminuisce la sensibilità del circuito ed elimina segnali con frequenze elevate che non sono necessari per questo esperimento.

Quando all'ingresso di questo circuito arriva un segnale captato dall'antenna, tale che possa essere interpretato come un impulso, cambia il livello dell'uscita della porta U1A, e quindi cambia anche il livello di uscita della porta successiva, e quando si genera un fron-

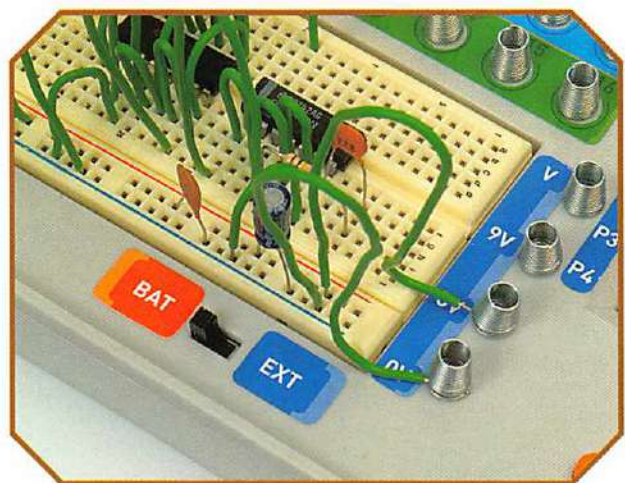


*Collegamenti all'ingresso del clock del contatore.*

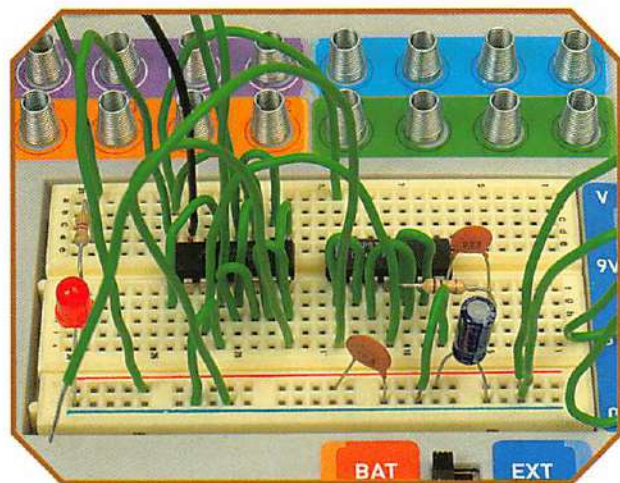
te di salita viene interpretato come un impulso di clock applicato all'ingresso del bistabile T formato con il bistabile JK del 4027.

La frequenza a 50 Hz è sufficientemente elevata perché l'occhio umano non ne percepisca la variazione, e se si utilizza un segnale a questa frequenza per illuminare un LED, la sensazione ottica ottenuta è che sia sempre illuminato.

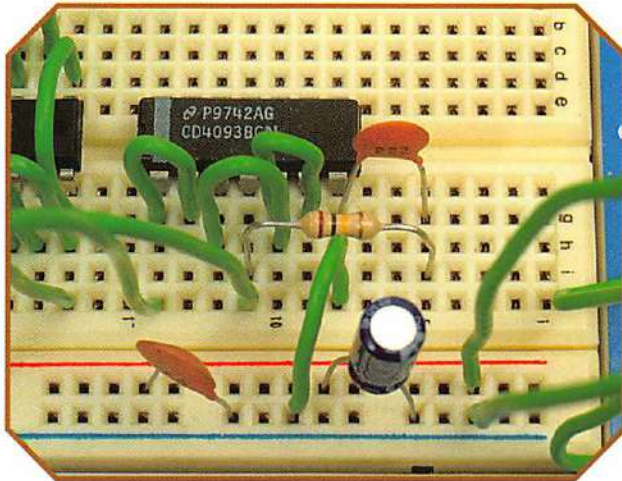
Facendo attraversare il segnale che abbiamo captato a questi due bistabili, come risultato otteniamo quello di dividere per quattro la frequenza del segnale stesso che, nel caso di 50 Hz, si riduce a 12,5 Hz e questa uscita è applicata all'ingresso del clock del contatore, terminale 1 del connettore J31 della scheda DG03, facendo così contare il contatore. Per visualizzare meglio l'effetto aggiungiamo un LED che si



*Collegamenti di alimentazione.*



*Un pezzo di filo capta il rumore.*

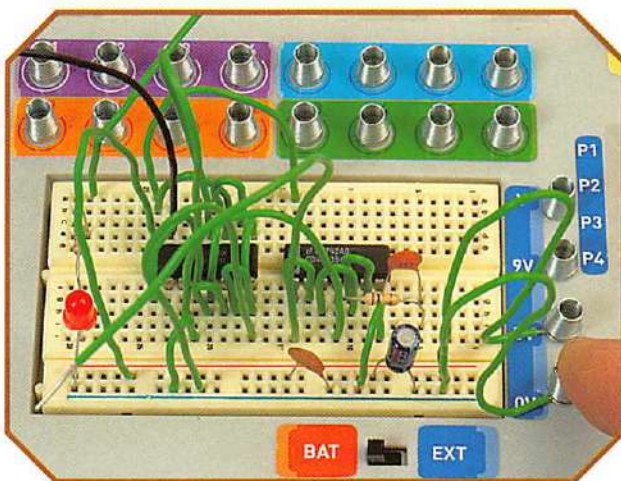


*Il condensatore C3 e la resistenza R1 diminuiscono la sensibilità del circuito.*

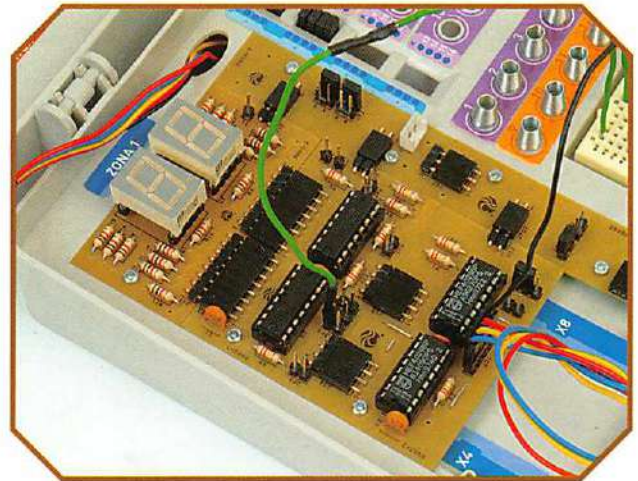
illumina quando il bit più significativo del primo display è a livello alto. Questo collegamento è disponibile sul terminale JP4, in modo che sia facile vedere il lampeggio del LED.

## Montaggio

Il montaggio si esegue come d'abitudine e si inizia collocando i componenti sulla scheda Bread Board, tenendo presente l'orientamento dei circuiti integrati, quello del condensatore elettrolitico C3 e del LED. Inoltre non ci dobbiamo dimenticare delle alimentazioni dei circuiti integrati riportate nella zona in basso a sinistra dello schema. L'antenna è un filo di una lunghezza qualsiasi il cui effetto potrà essere apprezzato realizzando l'esperimento.



*Dobbiamo scaricare l'elettricità statica toccando il terminale a 0 V.*

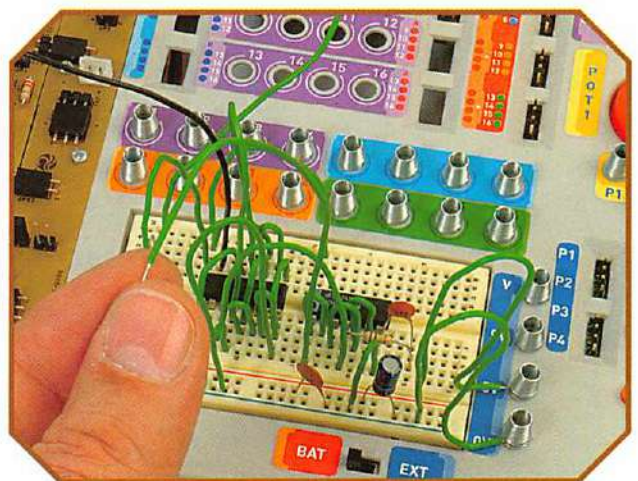


*Il contatore deve essere alimentato alla stessa tensione degli integrati che si trovano sulla scheda Bread Board.*

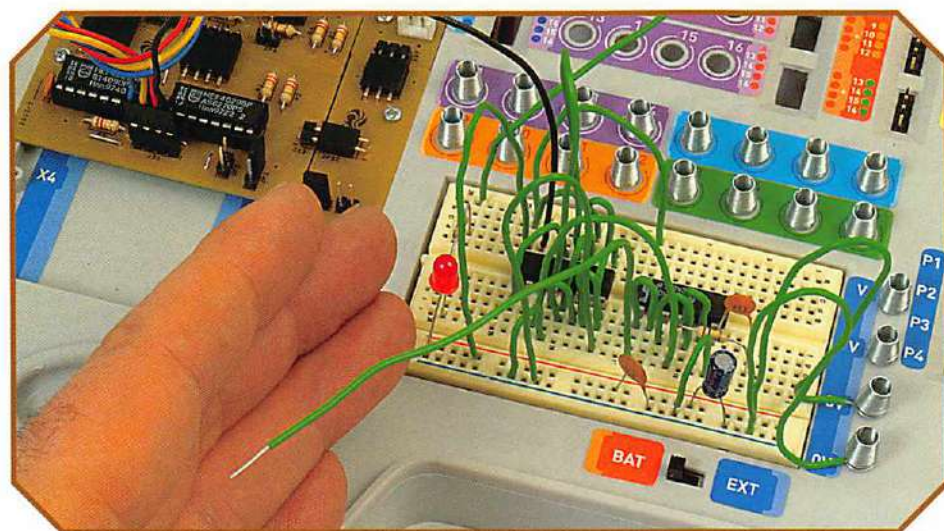
Il ponticello JP2 della scheda DG03 deve essere inserito per fare in modo che il contatore conti le decine. I ponticelli dell'alimentazione delle schede DG04 e DG05 devono essere posizionati per 5 o 9 V. Questo circuito si può alimentare a 5 o 9 V, però le schede del contatore devono essere alimentate anch'esse alla stessa tensione utilizzata per i circuiti integrati montati sulla scheda Bread Board.

## Funzionamento e precauzioni

In alcuni casi questo esperimento potrebbe sembrare difficile da mettere in funzione, tuttavia non dobbiamo dimenticare che i circuiti che progettiamo devono sopportare le condizioni ambientali reali e, anche se in questo



*Dopo aver scaricato l'elettricità statica, possiamo toccare la punta dell'antenna con il dito.*



*Se il disturbo è forte, può essere sufficiente avvicinare la mano.*

esperimento sarà necessario collegare un filo di una certa lunghezza: 20 cm, 1 m, 2 m, per enfatizzare il malfunzionamento e la captazione del rumore, in molti circuiti reali è possibile captare il rumore con estrema facilità. In alcuni casi potremo vedere che sarà sufficiente avvicinare la mano al laboratorio per far rilevare il rumore, illuminando il LED e facendo contare il contatore.

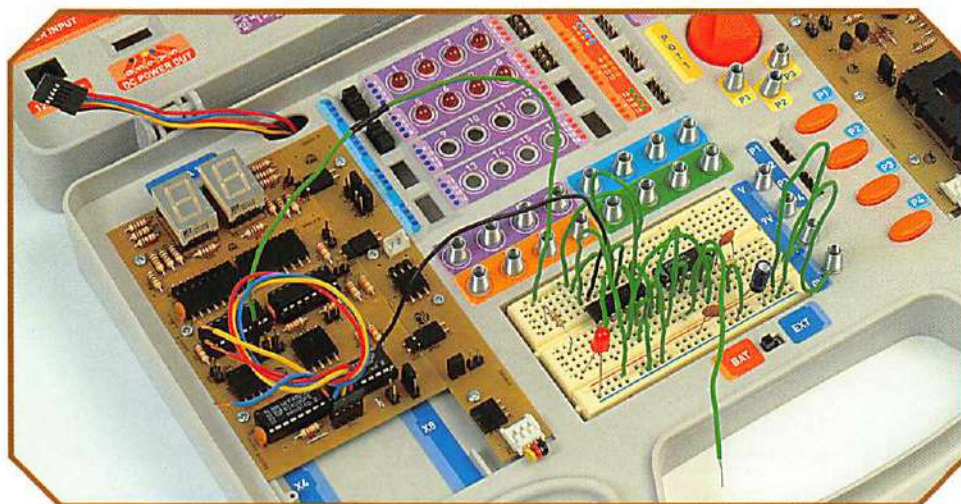
Se vogliamo toccare con un dito il filo utilizzato come antenna, è necessario prima toccare con lo stesso dito la molla siglata come 0 V per scaricare l'elettricità statica, dato che se siamo molto caricati potremmo danneggiare il circuito integrato.

Se avviciniamo un filo alla rete di alimentazione di qualche strumento, dovremo fare molta attenzione a non toccarlo e a utilizzare un fi-

lo isolato (non bisogna mai collegare direttamente alla rete); potremo vedere che il circuito inizierà a contare e il LED a lampeggiare. Il circuito può anche funzionare quando si tocca con la mano l'involucro esterno del filo utilizzato come antenna e con l'altra il cavo dell'alimentazione, ad esempio di una lampada.

#### LISTA DEI MATERIALI

U1	Circuito integrato 4093
U2	Circuito integrato 4027
R1	Resistenza 100 K (marrone, nero, giallo)
R2	Resistenza 4K7 (giallo, viola, rosso)
C1,C3	Condensatore 100 nF
C2	Condensatore 10 µF elettrolitico
LEDA	Diode LED rosso 5 mm



*Vista di un particolare del laboratorio con l'esperimento completato.*





## Esercizi 10 e 11: la pratica

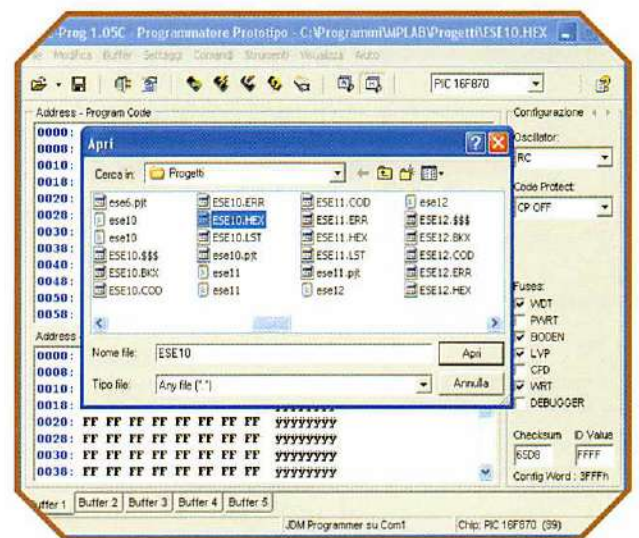
**D**opo aver verificato che gli esercizi 10 e 11 vengono compilati correttamente, si esegue la simulazione. Il passaggio finale è quello di scrivere il microcontroller ed eseguire il montaggio dell'hardware necessario perché possa funzionare realmente sul laboratorio.

### Scrittura del microcontroller

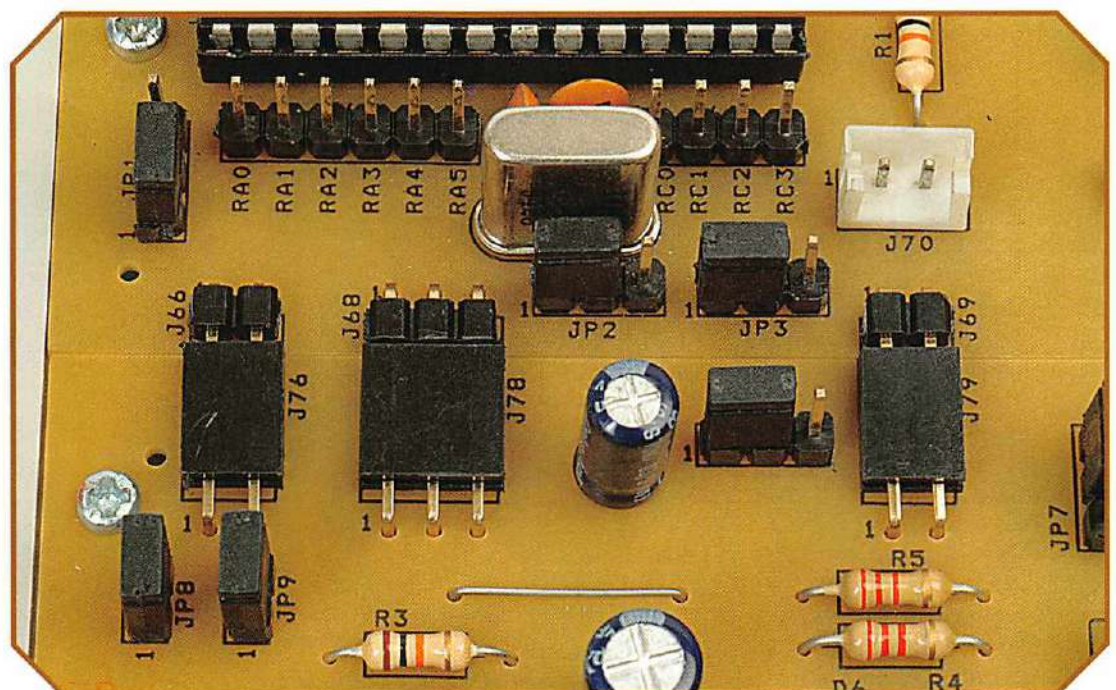
Per scrivere qualsiasi programma sul microcontroller dobbiamo collegare il laboratorio al PC e utilizzare il software IC-Prog. A questo scopo non abbiamo bisogno di alimentare il laboratorio.

Per rendere effettivo il trasferimento dei dati è necessario configurare l'hardware in modo adeguato. Ricordate che per poter utilizzare l'alimentazione della porta seriale del PC, i ponticelli dei connettori JP1, JP2 e JP3 devono essere nella posizione 1 e 2 e per poter trasferire dati al PIC i ponticelli dei connettori JP8 e JP9 devono essere inseriti.

Eseguita la configurazione hardware, avvieremo IC-Prog e interagirremo col microcontroller. Vi consigliamo di leggere il contenuto del



Aprire il file .hex che si desidera caricare sul PIC.



Ogni volta che si vuole scrivere sul PIC sarà necessario avere la stessa configurazione dei ponticelli.



Configurazione

Oscillatore:  
XT

Code Protect:  
CP OFF

Fuses:

WDT  
 PWRT  
 BODEN  
 LVP  
 CPD  
 VRT  
 DEBUGGER

Checksum	ID Value
630E	FFFF

Config Word: 3D35h

Configurazione di IC-Prog per la scrittura.

dispositivo prima di effettuare un'operazione di cancellazione o di scrittura, nel caso il microcontroller contenesse dei programmi che vogliamo conservare. Potremo quindi cancellare il PIC, leggendone nuovamente il contenuto per verificare la fase di cancellazione. Sarà necessario aprire il file che si desidera scrivere (estensione ".hex") e configurare l'oscillatore e i bit della parola di configurazione, supponendo che sia il dispositivo che la protezione del codice siano correttamente selezionati (PIC16F870 e CP OFF). Attivate le opzioni WDT e PWRT della parola di configurazione e procedete a scrivere il PIC, verificando il processo con una lettura successiva del dispositivo.

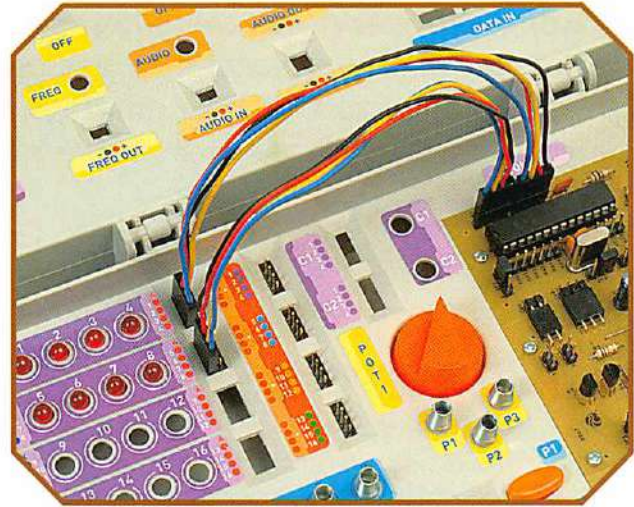
Il processo di scrittura è sempre lo stesso per tutti i programmi, con qualche differenza per i programmi che richiedono l'attivazione di qualche altro bit della parola di configurazione in funzione dei dispositivi utilizzati, come, ad esempio, la memoria FLASH.

## Esercizio 10, la pratica

Caricate l'esercizio 10 sul microcontroller. Questo esercizio provoca l'accensione successiva della barra dei LED ogni 1,3 s, simulando un contatore binario.

Per procedere al montaggio dell'applicazione, fase finale di qualsiasi progetto, ricordate che sarà necessario modificare i ponticelli del laboratorio. I connettori JP1, JP2 e JP3 dovranno avere i ponticelli tra i terminali 2 e 3 per ricevere l'alimentazione dal laboratorio stesso e i connettori JP8 e JP9 dovranno avere i ponticelli estratti.

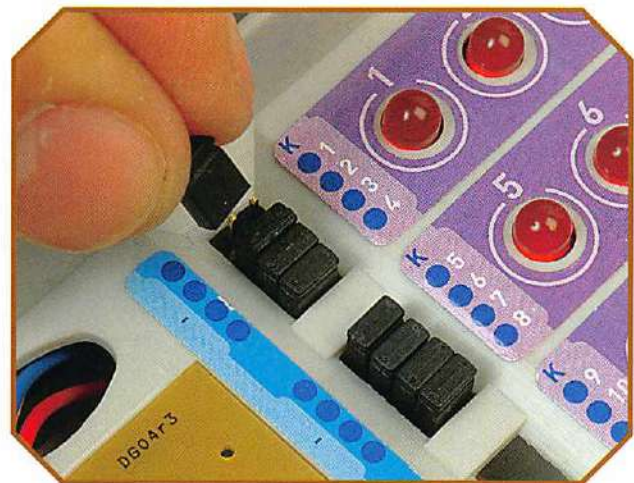
Nell'esercizio si lavora con la matrice dei LED, che si controlla tramite la porta B del microcontroller. Ricordate che quando si lavora con la matrice dei LED è necessario impostare i ponticelli sui catodi dei diodi LED, come si può vede-



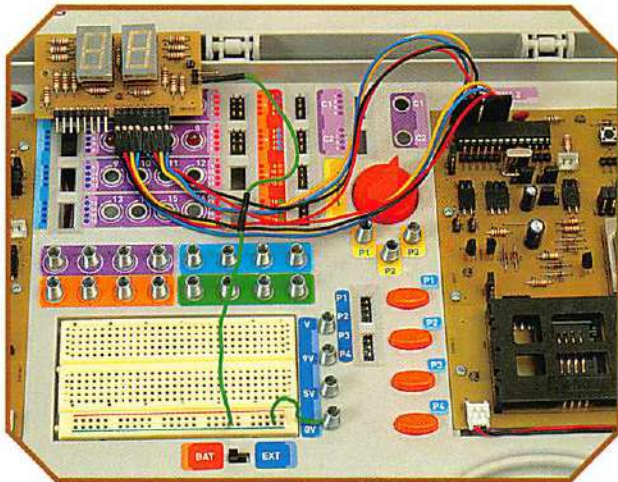
Collegiamo i terminali della porta B con la matrice dei LED.

re nella figura a fianco. Dobbiamo collegare la porta B con la matrice dei LED tramite due dei cavetti di collegamento a quattro fili. In questo modo uniremo con uno dei cavetti i terminali da RB0 a RB3 con la prima fila dei diodi, e con l'altro i terminali da RB4 a RB7 con la seconda fila della matrice.

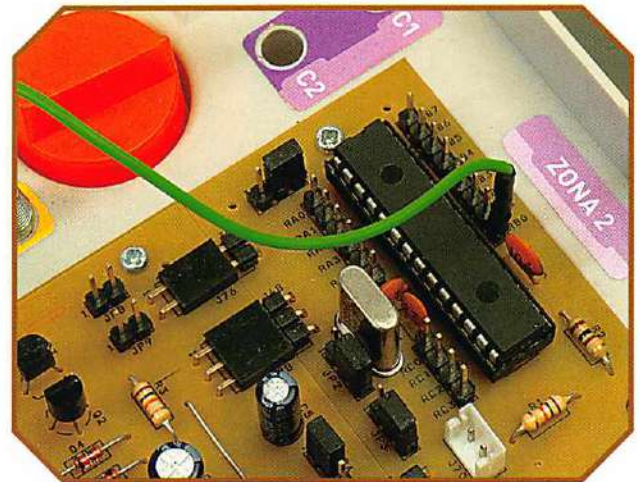
Questo montaggio è identico a quello realizzato per l'esercizio 9 in cui si lavorava con il TMR1. Nelle immagini che rappresentavano il montaggio di quella applicazione si utilizzava solamente un cavetto, che collegava i terminali RB0:RB3 con la fila superiore della matrice dei LED, mentre nell'esercizio presente ne stiamo utilizzando due per unire tutta la porta B con la matrice dei LED. Possiamo verificare come i LED



Dobbiamo inserire i ponticelli sui catodi dei diodi LED.



Collegiamo l'uscita al display a 7 segmenti.



Collegiamo un filo all'unico terminale di uscita che vogliamo utilizzare (R0).

si accendano in modo sequenziale simulando un contatore binario.

Questo esercizio può anche essere eseguito utilizzando i LED interni a uno dei display. A questo scopo si collega uno dei display a 7 segmenti invece della matrice dei LED, osserveremo così lo stesso risultato ma su un altro dispositivo di visualizzazione. Ricordate che per collegare i display dovremo staccare la scheda DG01 e avvicinarla alla porta B, facendo sempre attenzione a non creare contatti con qualche oggetto metallico. Per quanto riguarda la tensione di riferimento sarà necessario unire il terminale 1 del contatore J13 con 0 V mediante un filo di collegamento. Nell'immagine si

può vedere il montaggio con il collegamento tra la porta B e il display a 7 segmenti.

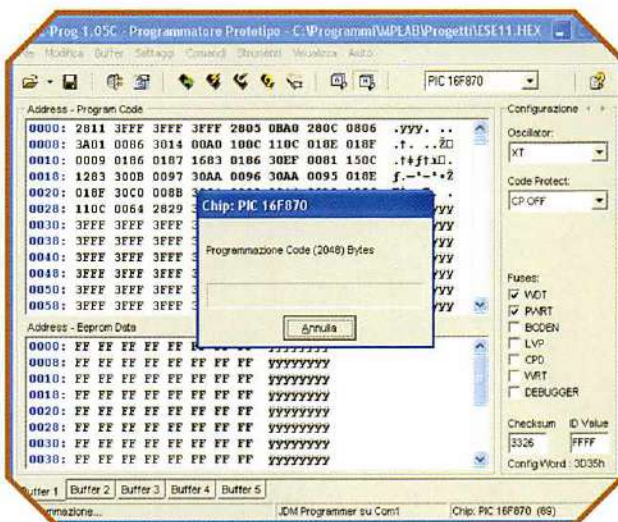
### Esercizio 11, montaggio

Questo esercizio ha come obiettivo quello di accendere o spegnere il diodo LED collegato a R0 quando il valore del TMR1 abbia coinciso per 20 volte con un valore precedentemente impostato sui registri CCPR1H/L del modulo CCP.

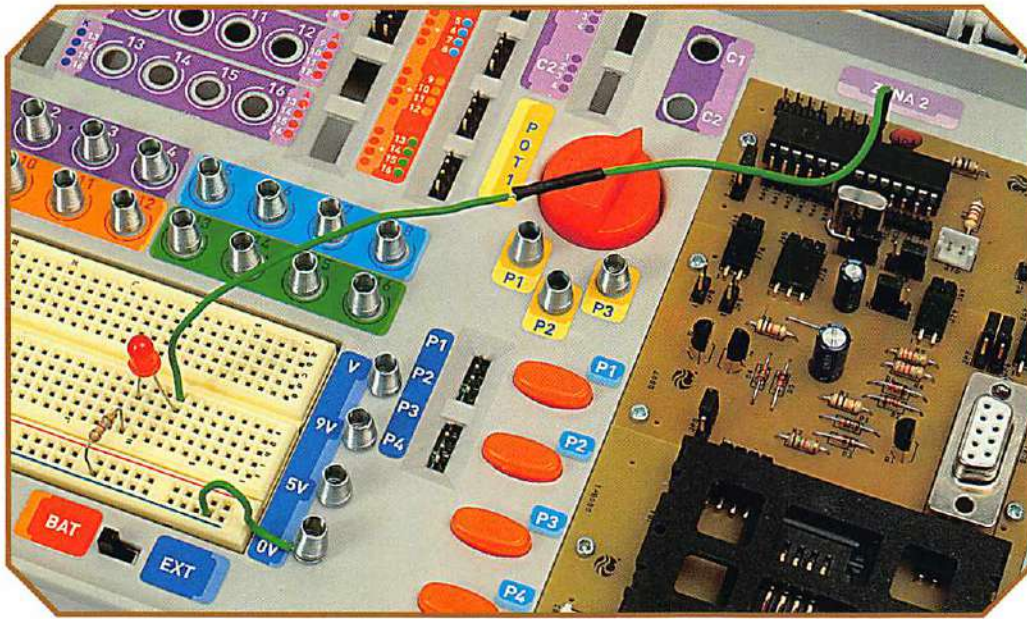
Sarà necessario configurare nuovamente il laboratorio in modo che i ponticelli siano posti in modo scrittura (JP1, JP2, JP3, JP8 e JP9). Collegate il cavo di comunicazione tra il PC e il laboratorio e fate partire il software IC-Prog. Abituatevi a eseguire sempre gli stessi passaggi: lettura del dispositivo, cancellazione e verifica. Caricate l'esercizio 11 e realizzate la configurazione del software prima di procedere alla scrittura. Cambiate il tipo di oscillatore e il bit di configurazione e programmate il dispositivo.

Dopo aver verificato che il programma sia stato caricato sul PIC potete procedere a riconfigurare il laboratorio, spostando i ponticelli sui connettori JP1, JP2 e JP3, e togliendo i ponticelli di JP8 e JP9.

Il montaggio di questa applicazione risulta più semplice dell'esercizio precedente. In questo caso, infatti, utilizzeremo unicamente un terminale della porta B come uscita, cioè R0. Dato che si tratta di accendere e spegnere un diodo LED, possiamo fare un semplice montaggio sulla scheda Bread Board in cui porremo una resistenza collegata a massa in serie al di-



Carichiamo l'esercizio 11 sul PIC.



Realizziamo il montaggio sulla scheda Bread Board.

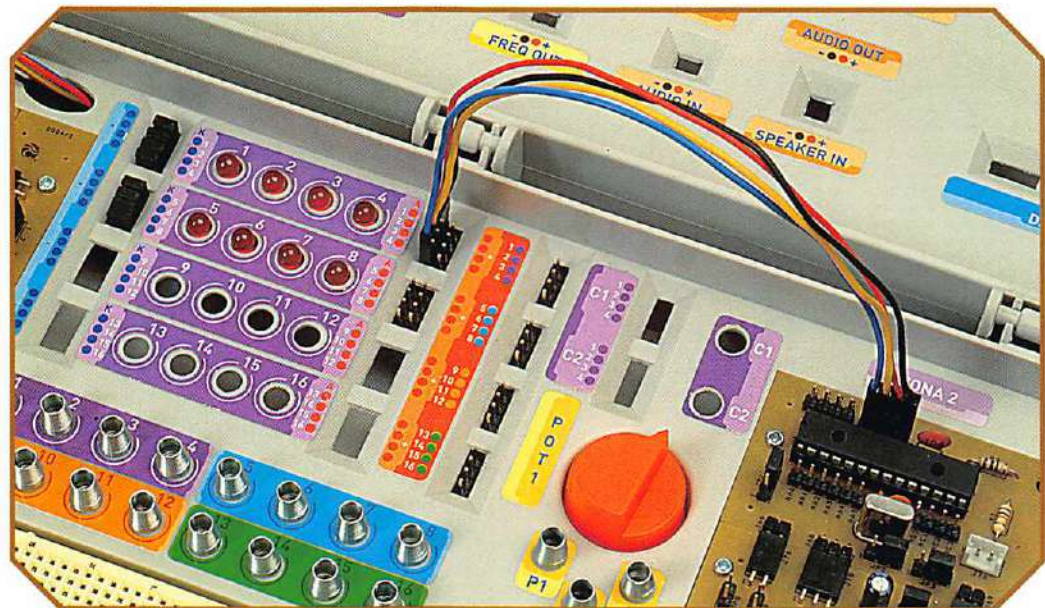
do, che a sua volta va collegato al terminale RB0. Nella figura possiamo vedere questo semplice montaggio.

Un'altra possibilità di realizzare l'applicazione sul laboratorio è utilizzando la matrice dei diodi. Anche se utilizzeremo solamente uno dei diodi, uniremo mediante un cavo di collegamento i terminali RB0:RB3 della porta B con la matrice dei diodi (i ponticelli sui catodi devono essere inseriti).

Come potremo verificare il diodo si accende e si spegne costantemente.

### Consigli

Potete vedere come vengono presentate diverse alternative di montaggio per poter in questo modo interagire con i diversi elementi del laboratorio. È altresì interessante trovare le diverse alternative ai codici previsti. Provate come esempio a variare il valore di comparazione di quest'ultimo esercizio e verificate che se si cambia il valore dei registri di comparazione CCPR1H e CCPR1L, varia la frequenza di accensione del diodo LED.



Possiamo anche collegare la matrice dei LED.



# Capacimetro sperimentale

**G**razie a questo circuito potremo provare a mettere in pratica l'idea di costruire un semplice strumento, basato su un contatore che utilizzato secondo un preciso criterio misura la capacità di un condensatore.

## Il circuito

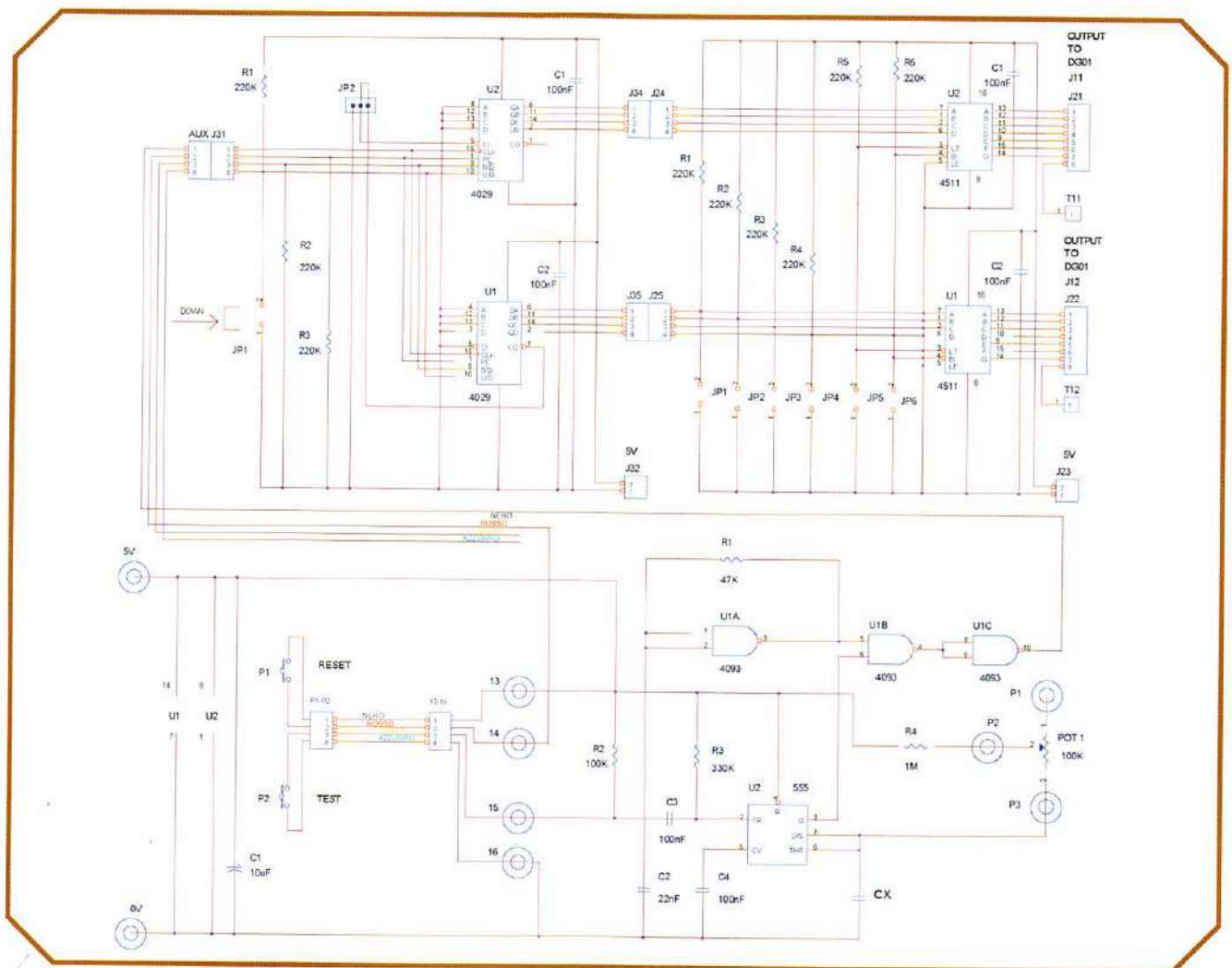
Questo circuito è piuttosto rudimentale e manca di elementi di regolazione e correzione, si può utilizzare solamente con lo scopo di dimostrare il principio su cui si basa.

Un circuito oscillatore applica degli impulsi all'ingresso del clock di un contatore per una "finestra" di tempo, e questa "finestra" di tempo dipende dalla capacità di misurare.

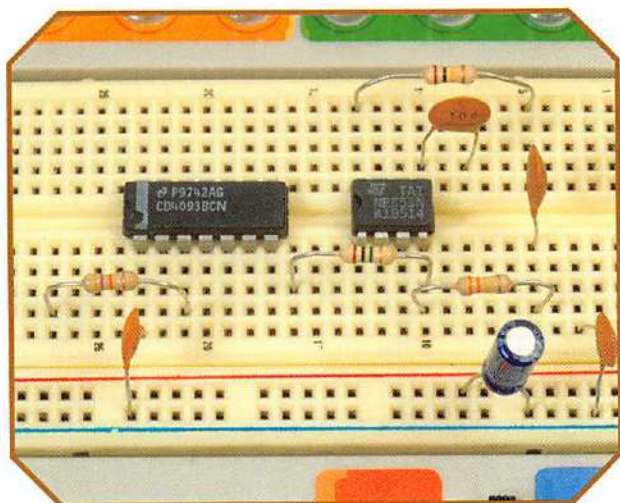
Per realizzare un circuito con un basso numero di componenti, limiteremo la misura a condensatori tra 10 e 99 nF.

## Lo schema

Osservando lo schema, vediamo che la porta U1A è un oscillatore astabile, con una frequenza teorica di 1 kHz. A questa frequenza lo dovremo regolare variando il valore di R1



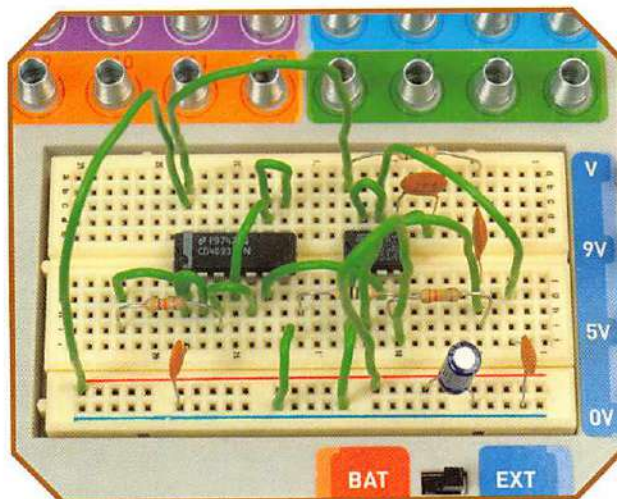
Schema elettrico.



Scheda Bread Board con i componenti.

per ottenere risultati precisi, l'uscita dell'oscillografo si applicherà alla porta U1B, questa porta è quella che permette o impedisce agli impulsi generati di arrivare al contatore. Questo controllo si realizza grazie a un monostabile formato da un circuito integrato 555, U2 nello schema. Il tempo in cui questa porta è "aperta" dipende dalla durata dell'impulso del monostabile, in altre parole dal tempo in cui un livello alto verrà applicato al terminale 6, ingresso di U1B.

Questo tempo è proporzionale alla capacità del condensatore, collegato al riferimento CX dello schema.

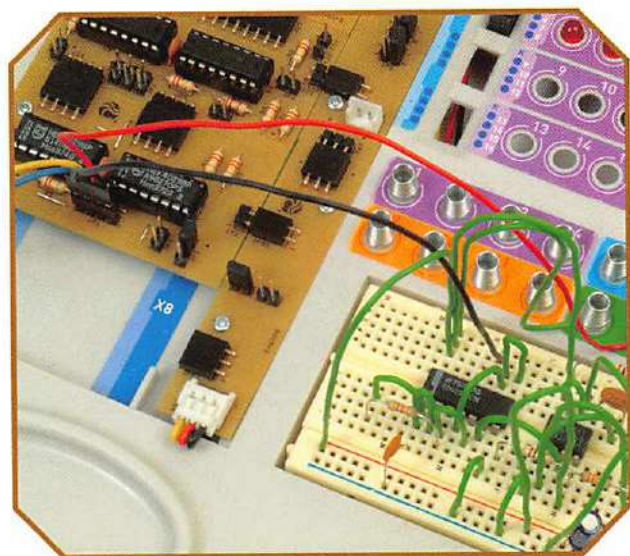


Cablaggio interno della scheda.

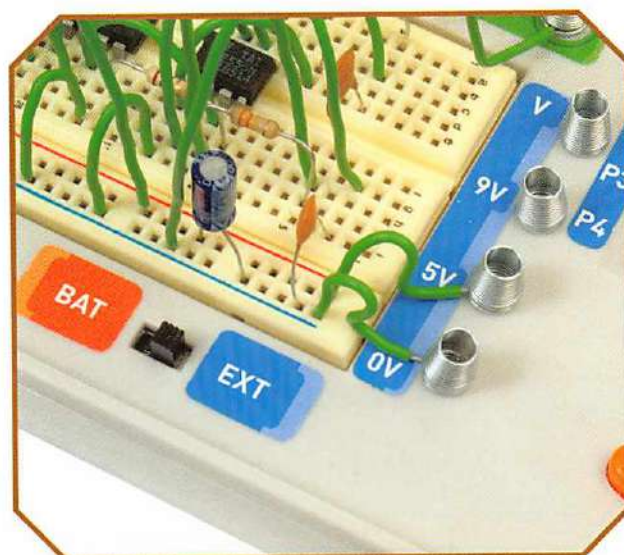
L'idea è di ottenere un tempo del monostabile di tanti millisecondi quanti sono i nanofarad da misurare. Questo lo si può ottenere con il potenziometro di regolazione, nel caso in cui il margine di regolazione di POT1 non fosse sufficiente, sarà necessario cambiare il valore di R4. In realtà l'obiettivo non è costruire uno strumento di misura di precisione, ma verificare il suo principio di funzionamento.

## Montaggio

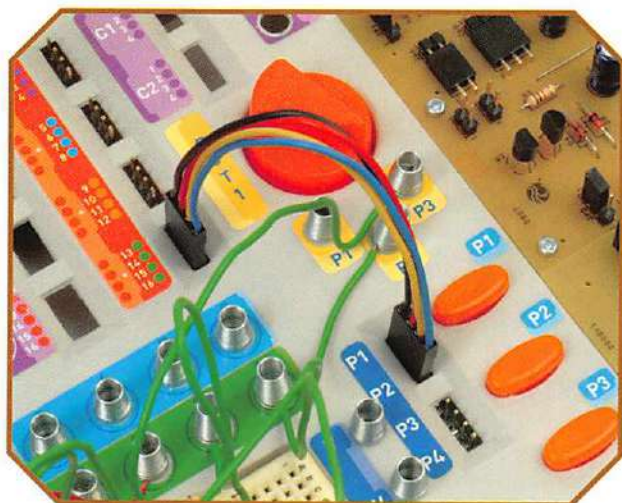
Il montaggio si esegue come d'abitudine, osservando lo schema e lavorando con pazienza.



Collegamenti a J31 di DG03.

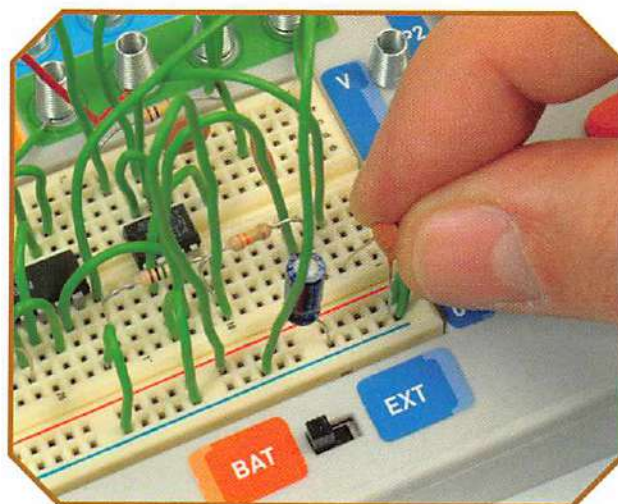


Collegamenti di alimentazione.



Collegamento ai pulsanti di RESET e TEST.

za per non commettere errori. Inizieremo dal montaggio dei componenti sulla scheda Bread Board, tenendo presente l'orientamento dei circuiti integrati e quello del condensatore elettrolitico C1. Il collegamento all'ingresso del condensatore si esegue con un cavetto terminato su un connettore a quattro vie, mentre il collegamento dei pulsanti si realizza grazie alle molle dalla 13 alla 16, utilizzando un cavetto a quattro fili terminato da entrambe le parti su dei connettori da quattro vie. È necessario collegare anche le alimentazioni dei circuiti integrati che sono indicate nella zona sinistra dello sche-



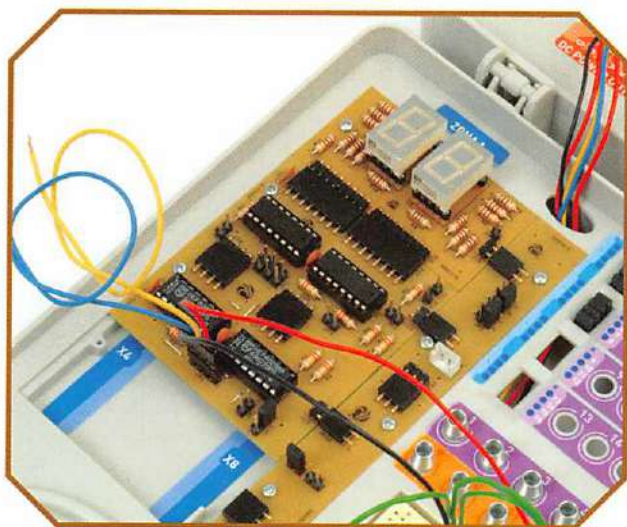
Il condensatore da misurare si collega nella posizione CX.

ma, per fare in modo che il contatore conti le decine.

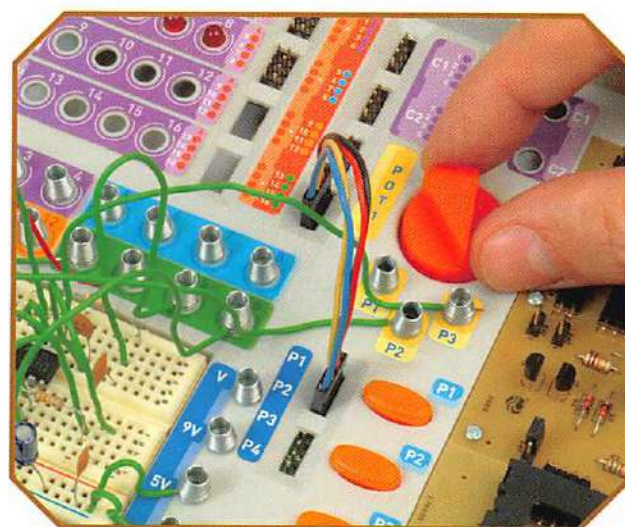
## Funzionamento

Dopo aver verificato che tutto il montaggio sia stato realizzato correttamente, sarà possibile collegare l'alimentazione che potrà essere da 5 o 9 V, sia per il contatore che per il resto dei dispositivi, in ogni caso la stessa per tutto l'insieme.

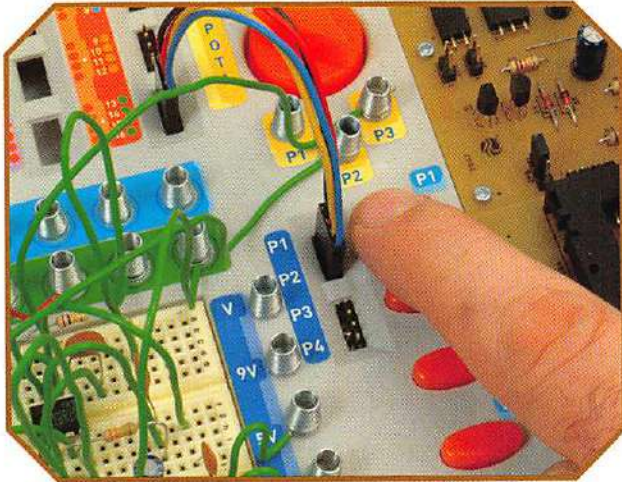
Con il pulsante P1 si imposta il contatore a 0 e premendo P2 e rilasciandolo inizia la misura e il contatore si fermerà su di un valore,



Il valore della capacità misurata si leggerà sul contatore.



Con il POT1 si regola la lettura.



Il pulsante P1 si utilizza come RESET.

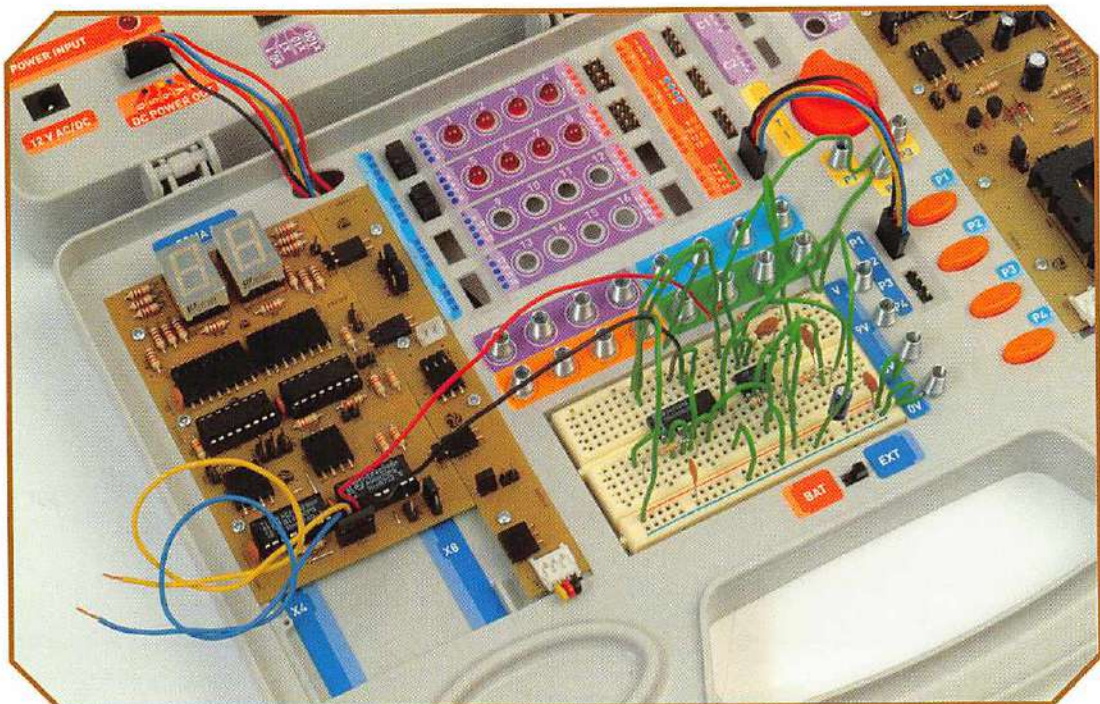


Il pulsante P2 indicato nello schema come TEST, si utilizza per eseguire la misura.

si monta un condensatore da 22 nF, ad esempio, nella posizione CX e si regola con POT1 fino a quando la lettura sul contatore sarà 22. Ricordiamo che non si tratta di uno strumento di precisione, prima abbiamo parlato del principio di funzionamento e di come si potrebbe ottenere un funzionamento più preciso se si potesse disporre di una strumentazione da utilizzare per le misure di riferimento.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
U2	Circuito integrato 555
R1	Resistenza 47 K (giallo, viola, arancio)
R2	Resistenza 100 K (marrone, nero, giallo)
R3	Resistenza 330 K (arancio, arancio, giallo)
R4	Resistenza 1 M (marrone, nero, verde)
C1	Condensatore 10 $\mu$ F elettrolitico
C2	Condensatore 22 nF
C3, C4	Condensatore 100 nF



Laboratorio con l'esperienza completato.





# Esercizio 12: modulazione di ampiezza degli impulsi PWM, la pratica

*In questo esercizio realizzeremo un generatore di onda quadra con ampiezza di impulso variabile, modulata. Questo montaggio corrisponde all'esercizio 12 del modulo CCP in modo PWM.*

## Scrittura del programma

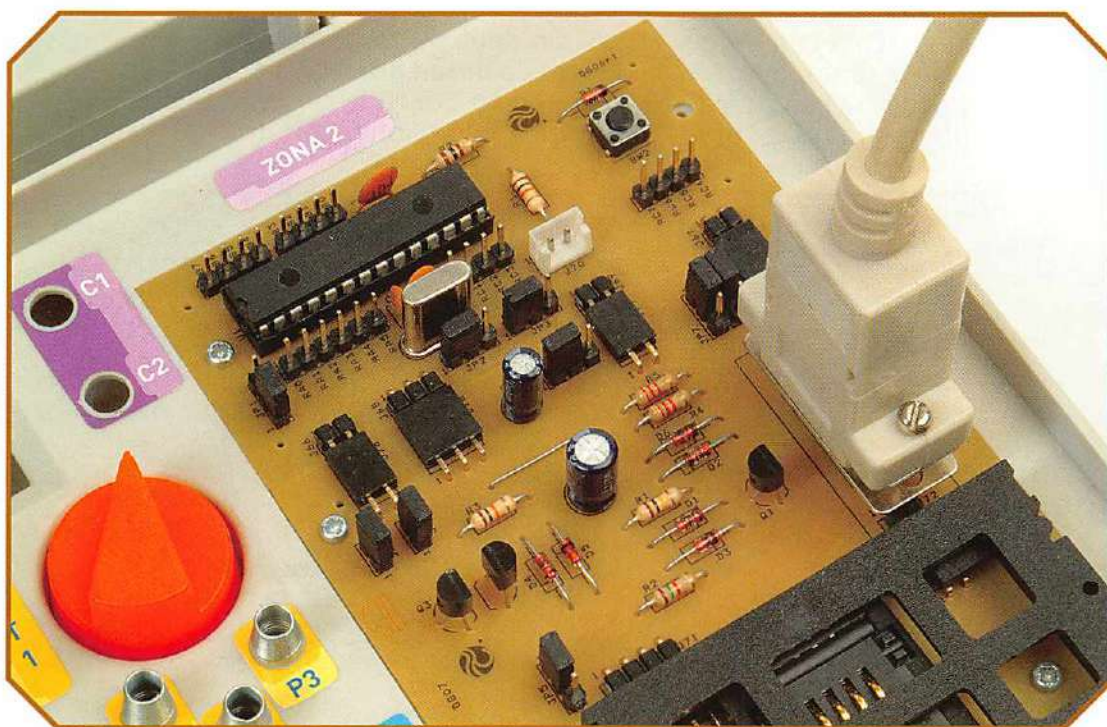
Dobbiamo caricare il file in codice macchina ".hex" sul nostro PIC, quindi prepareremo il laboratorio per lavorare con IC-PROG, per questo imposteremo i ponticelli sui connettori JP1, JP2 e JP3 sui PIN 1 e 2 e inseriremo i ponticelli sui connettori J8 e J9. Collegheremo poi il cavo di trasferimento o comunicazione fra il PC e il laboratorio e faremo partire il software di scrittura.

I passaggi che si devono eseguire per scrivere un programma sul microcontroller, sono sempre gli stessi e sono riportati nel digramma di flusso della pagina successiva. Leggeremo il dispositivo e lo cancelleremo, verificando di averlo cancellato correttamente. Fatto questo configureremo l'oscillatore (XT), la protezione del codice (CP OFF) e i bit della pa-

rola di configurazione (WDT e PWRT), saremo quindi pronti per scrivere, quindi cliccheremo l'icona "Programma tutto" sulla barra degli strumenti. In ultimo verificheremo se la scrittura è stata realizzata con successo, leggendo nuovamente il dispositivo e verificando che contenga il programma desiderato.

## Montaggio. Configurazione del laboratorio

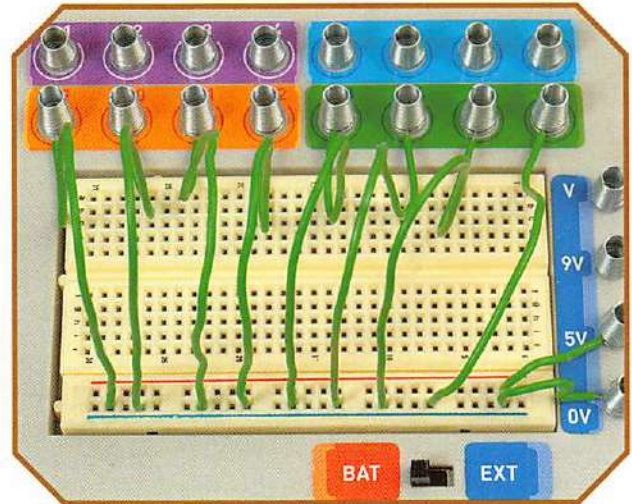
Con il programma caricato sul microcontroller dobbiamo configurare il laboratorio per il modo lavoro normale, in altre parole scollegheremo il cavo di commutazione, prenderemo l'alimentazione dalle batterie spostando i ponticelli dei connettori JP1, JP2 e JP3 sulle posizioni 2 e 3, e toglieremo i ponticelli dei connettori JP8 e JP9.



Configurazione hardware del laboratorio in modo scrittura.



Sequenza da seguire per scrivere il programma sul microcontroller.



Collegamenti degli ingressi sulla scheda.

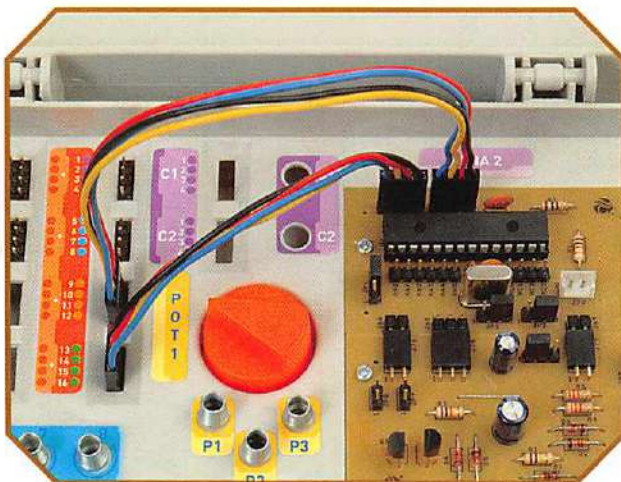
## Ingressi

Possiamo modificare l'ampiezza dell'impulso (Duty Cycle) in funzione degli ingressi digitali della porta B, dobbiamo quindi fare arrivare a questa porta diverse combinazioni di stati logici. A questo scopo collegheremo alle molle di alimentazione da 5 V un filo che all'altro estremo andrà a una delle file di collegamento della scheda Bread Board e con un altro filo uniremo la molla da 0 V all'altra fila della scheda. Utilizzeremo le molle di collegamento dalla 9 alla 16 (blocco arancio e verde) per fare arrivare alla porta i segnali digitali, quindi a ognuna di queste molle collegheremo un filo

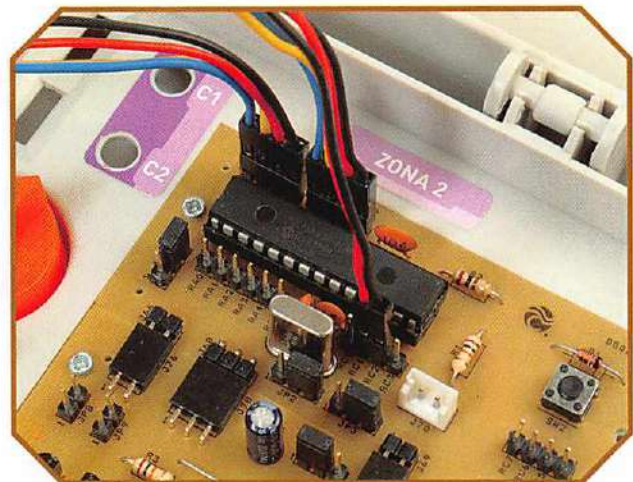
e l'altro estremo di quest'ultimo verrà collegato alla fila da 5 o da 0 V, in funzione della combinazione che desideriamo inserire. Inizialmente collegheremo tutti i fili alla linea 0 V, adottando il montaggio che possiamo vedere nella figura in alto.

Le molle sono collegate internamente con la fila di connettori situata a sinistra del potenziometro, quindi unendo mediante dei cavetti questi connettori con i due connettori della porta B, potremo fare arrivare a ognuno dei pin della porta, i valori logici presenti sulla scheda Bread Board.

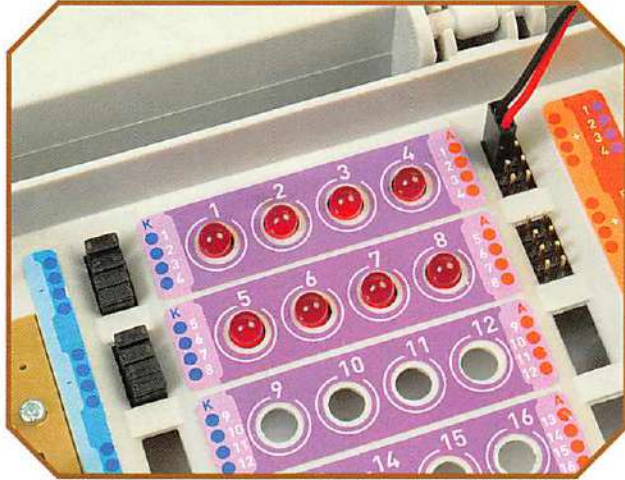
A questo punto la sezione degli ingressi di questa applicazione è pronta.



Portiamo gli ingressi al microcontroller tramite dei cavetti.



Collegiamo l'uscita mediante un cavetto a due fili.



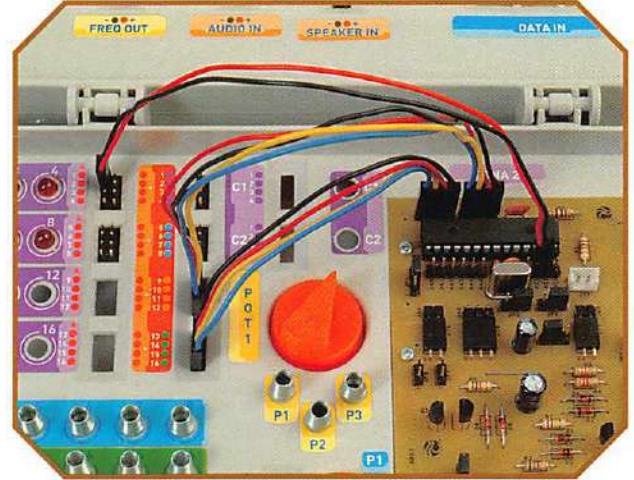
Portiamo l'uscita al diodo LED.

## Uscita

L'esercizio prevede una sola uscita e da essa otterremo l'onda quadra risultante dalla modulazione. Questa uscita è la RC2/CCP1. Collegheremo un cavetto a due fili al connettore della porta C su cui sono presenti i terminali RC0:RC3, in modo che uno dei due fili corrisponda al terminale RC2. Nella figura possiamo vedere questo collegamento.

L'altro estremo del cavetto lo collegheremo alla matrice dei diodi, in modo che mediante uno di essi potremo vedere come risponde all'uscita, cioè come si visualizza l'onda quadra.

Per lavorare con la matrice dei diodi LED, o

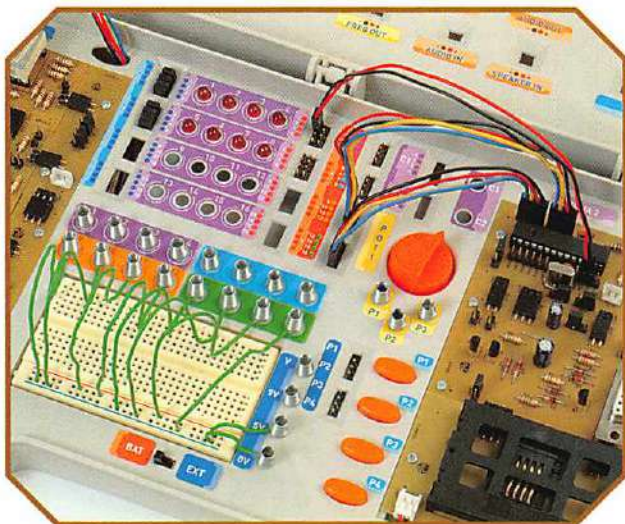


Montaggio degli ingressi e uscite.

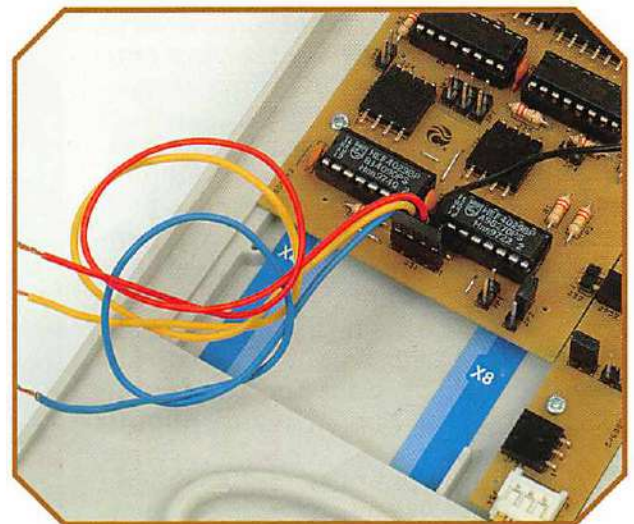
con qualche diodo di essa, è necessario inserire i ponticelli sui catodi dei LED che vogliamo utilizzare.

## Funzionamento

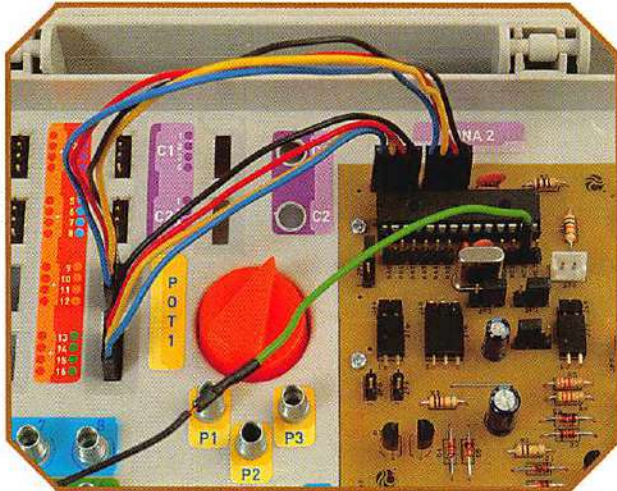
Dopo aver eseguito il montaggio completo degli ingressi e delle uscite, dobbiamo provare il funzionamento del modulatore. Se teniamo la porta B a "0" il LED resterà spento, mentre con tutta la porta a "1" il LED rimarrà sempre acceso. Inizialmente tutti i fili saranno "0" quindi non potremo vedere nulla sull'uscita. Se l'impulso è molto stretto il LED non si accende, ma se è un po' più ampio inizia ad ac-



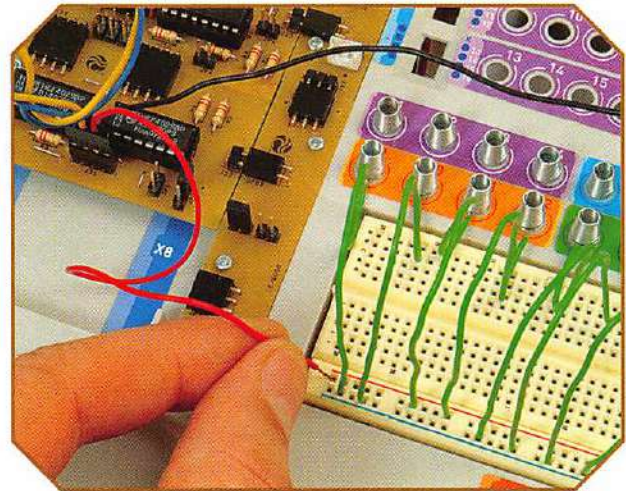
Aspetto del laboratorio durante il funzionamento della prova.



Colleghiamo un cavetto al contatore del display.



Collegiamo il filo dal contatore al terminale RC2.



Impostiamo il contatore a 0.

cendersi aumentando la propria luminosità. Per poter vedere questo passeremo alcuni degli ingressi a "1", inserendo i fili sulla fila da 5 V della scheda Bread Board.

Dato che la frequenza è molto alta non potremo apprezzare il lampeggio del diodo, ma se cambiamo gli ingressi, cioè se variamo l'ampiezza dell'impulso dell'onda, noteremo delle differenze nell'intensità del diodo che si illuminerà più o meno in base all'ampiezza dell'impulso.

## Alternative di montaggio

Dopo aver provato diverse combinazioni di ingressi e aver visualizzato il risultato sul diodo, proviamo un montaggio diverso utilizzando il display a sette segmenti. Utilizziamo il contatore con il display per poter verificare che con qualsiasi configurazione della porta B, esso conterà rapidamente gli impulsi indipendentemente dalla loro ampiezza. Il contatore non è influenzato dall'ampiezza degli impulsi, dato che la loro frequenza è costante, ma nel caso che tutta la porta B sia a "0" o a "1" cesserà di contare e verrà visualizzato un numero sul display, dato che con queste configurazioni non arrivano impulsi sull'uscita RC2.

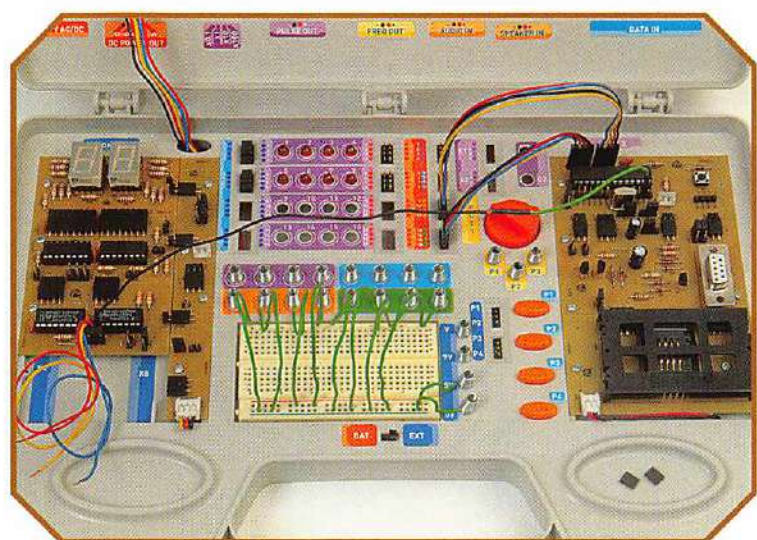
Sul connettore J31 della scheda DG03 collegheremo un cavetto a quattro fili, rispettando la posizione riportata nella figura (filo nero a destra). Il filo nero di questo cavetto lo dobbiamo

collegare all'uscita RC2 del microcontroller, dovremo pertanto utilizzare un filo intermedio per raggiungere la lunghezza adeguata.

Proveremo il funzionamento di questo nuovo montaggio e verificheremo che il contatore si incrementi a ogni impulso indipendentemente dal valore degli ingressi (sempre che non siano tutti a "0" oppure a "1").

Nell'immagine della figura possiamo vedere l'aspetto finale di questo esempio.

Se colleghiamo il cavo rosso del connettore J31 al positivo (5V) della scheda Bread Board, il contatore si imposterà a "0" indipendentemente dalla combinazione di ingressi che abbiamo configurato.



Aspetto del laboratorio dopo aver collegato il display all'uscita.



## Esercizio 13: convertitore A/D, la pratica

In questo esercizio eseguiamo il montaggio di un circuito invertitore analogico/digitale (CAD, ADC, A/D). Nel fascicolo precedente abbiamo realizzato un programma che convertiva un segnale analogico in uno digitale e lo rappresentava in binario mediante la matrice dei LED. Eseguiremo il montaggio di questa applicazione e ne proveremo il funzionamento.

### Scrittura del microcontroller

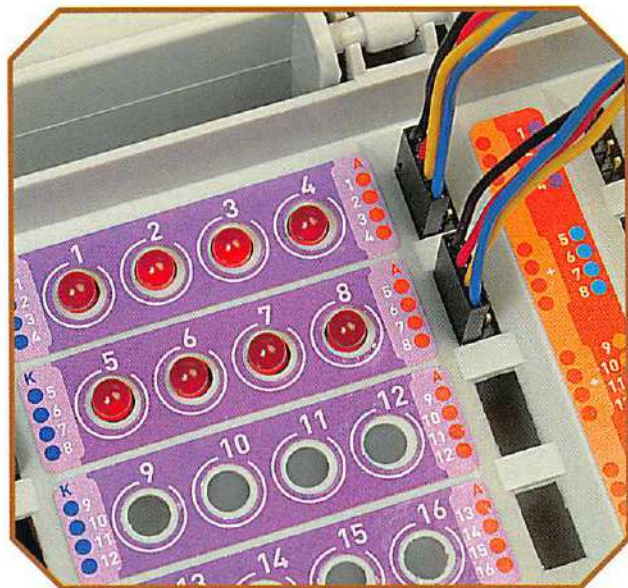
Quando compiliamo il codice con MPLAB otteniamo il file nel codice macchina corrispondente all'esercizio 13 (ese13.hex), ora dobbiamo scrivere questo file sul microcontroller. La prima cosa da fare sarà configurare l'hardware del nostro laboratorio. Verificheremo che il PIC sia inserito correttamente nel suo zoccolo, la presenza dei ponticelli JP8 e JP9 e che i con-

nettori JP1, JP2 e JP3 siano sulle posizioni 1 e 2. Collegheremo il laboratorio e il PC tramite il cavo di comunicazione, e faremo partire il software IC-Prog, a questo punto siamo pronti per eseguire il trasferimento.

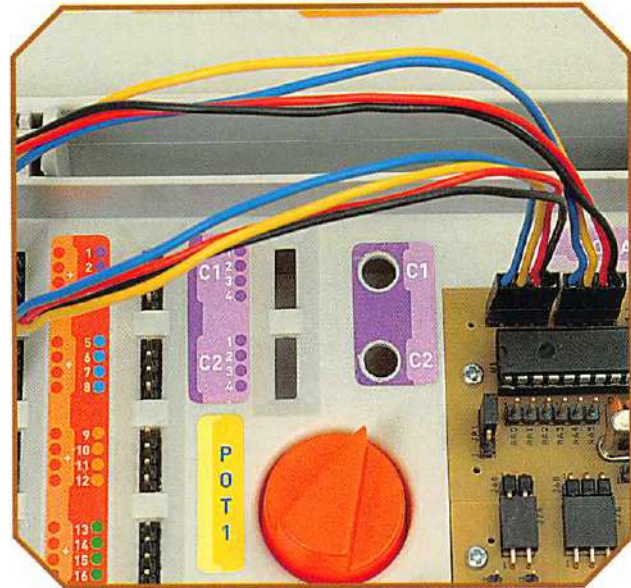
Ricordiamo i passaggi che bisogna seguire per scrivere un programma sul microcontroller: lettura, cancellazione, verifica, scrittura e verifica. Dopo aver eseguito le prime tre fasi, apriremo il file che vogliamo scrivere sul PIC e

The screenshot shows the IC-Prog 1.05C software interface. The main window displays the program code for a PIC16F870. The code is organized into two sections: 'Address - Program Code' and 'Address - Eeprom Data'. The program code section shows a list of addresses and their corresponding hex values, with some values being '3FFF' and others being '081E', '0086', '0008', etc. The Eeprom Data section shows a list of addresses and their corresponding hex values, all being 'FF'. The right side of the interface shows the 'Configurazione' (Configuration) panel, which includes settings for the oscillator (XT), code protection (CP OFF), and fuses (WDT, PWRRT, BODEN, LVP, CPD, WRT, DEBJGGER). The bottom of the interface shows the 'Buffer' selection (Buffer 1 to Buffer 5) and the connection information (JDM Programmer su Com1, Chip: PIC 16F870 (E9)).

Apriamo il file e configuriamo IC-Prog.



Collegiamo i cavetti alla matrice dei LED.



Collegiamo la matrice dei LED alle uscite del PIC.

configureremo l'oscillatore, la protezione del codice e i bit della parola di configurazione. Ora IC-Prog avrà l'aspetto della figura riportata nella pagina precedente.

Selezioneremo "Programma Tutto" e il programma si trasferirà al PIC. In ultimo, verifi-

cheremo che la scrittura sia stata eseguita correttamente leggendo il dispositivo.

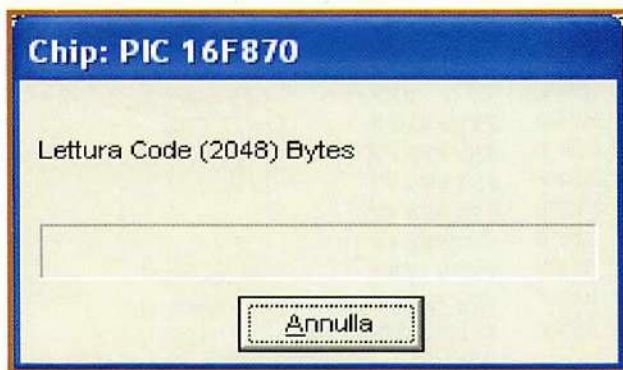
## Montaggio

### Configurazione del laboratorio

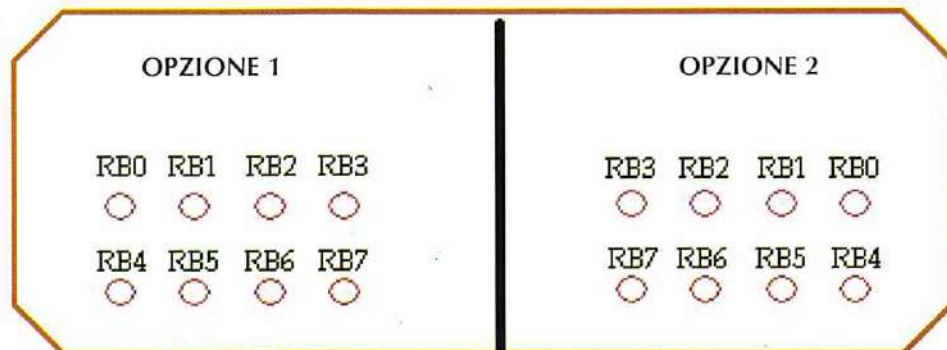
Dobbiamo preparare il laboratorio per provare il funzionamento dell'applicazione, per questo ripristineremo la configurazione precedente a quella richiesta per la scrittura. Scollegheremo il cavo di trasferimento, toglieremo i ponticelli dai connettori JP8 e JP9 e sposteremo sulle posizioni 2 e 3 i ponticelli dei connettori JP1, JP2 e JP3.

### Le uscite

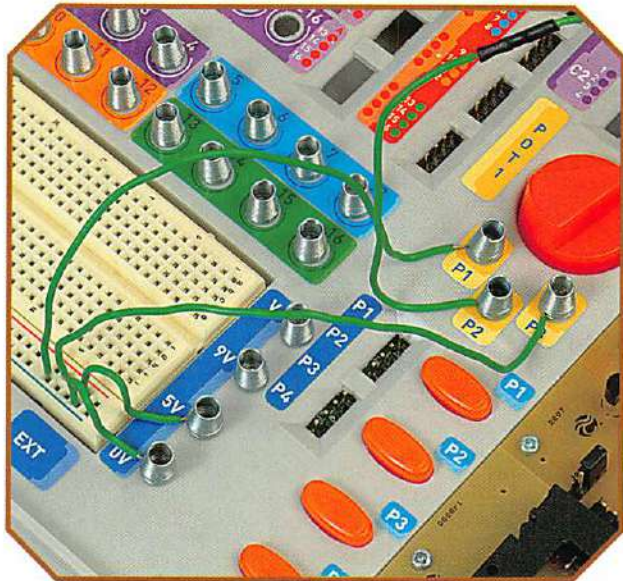
L'applicazione mostrerà, mediante la matrice dei LED, il risultato della conversione di un se-



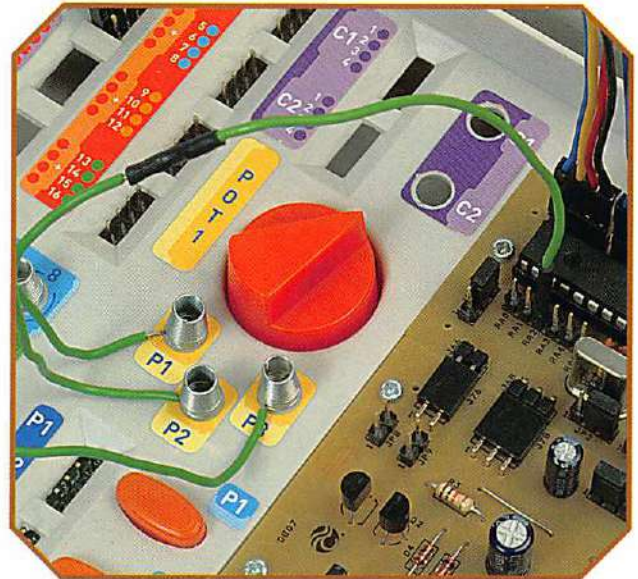
Leggiamo il PIC per verificare la fase di scrittura.



Due possibilità di visualizzare le uscite.



Collegiamo il potenziometro all'alimentazione.



Montaggio completo dell'ingresso analogico.

gnale analogico nel suo corrispondente digitale. Questo risultato verrà fornito in formato binario a 8 bit, avremo quindi bisogno di due file di LED. Dobbiamo inserire i ponticelli sui catodi, sulla parte sinistra della matrice, e collegare i due cavetti a quattro fili agli anodi. Possiamo vedere questo collegamento nelle immagini della pagina precedente.

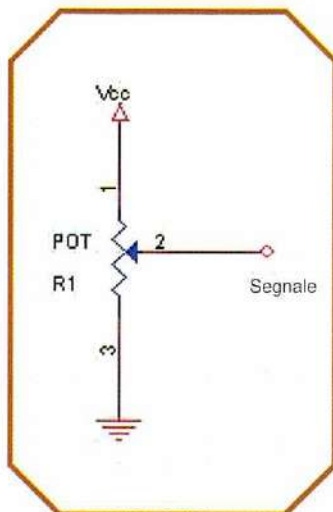
La porta del microcontroller che utilizzeremo come uscita è la porta B. La parte libera dei due cavetti verrà collegata a questa porta. Faremo corrispondere la fila superiore della matrice con i quattro bit meno significativi della

porta (RB3:RB0) e la fila inferiore con i quattro più significativi (RB7:RB4). Questo collegamento può essere fatto in due modi, come possiamo vedere anche dalla figura, dove nella prima opzione il diodo 1 corrisponde a RB0, il 2 a RB1 e così via; oppure possiamo incrociare i cavi in modo che il diodo 4 corrisponda con RB0, il 3 con RB1 ecc.. Quest'ultima rappresentazione dell'uscita sarebbe la più adatta a interpretare il risultato, in quanto lavorando con un formato binario i bit meno significativi rimarrebbero a destra.

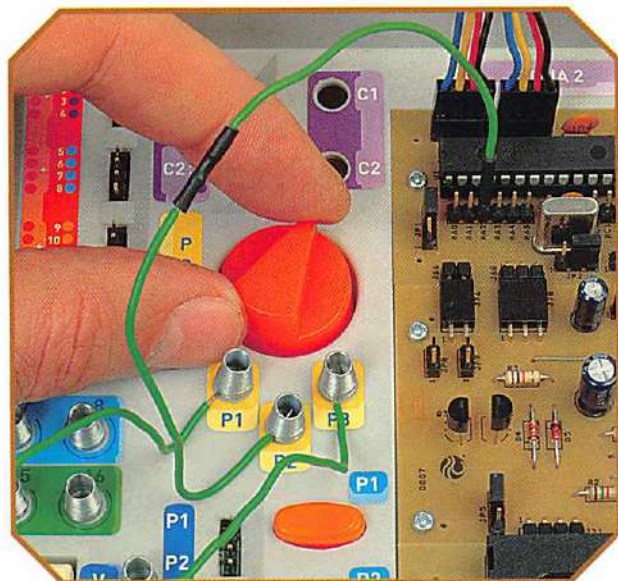
## L'ingresso

Il segnale di ingresso dell'applicazione è un segnale analogico che otteniamo dal potenziometro. Un potenziometro è una resistenza variabile, che risponde allo schema elettrico della figura. A uno dei capi della resistenza collegheremo l'alimentazione (5 Volt) e all'altro capo la massa o 0 V. In funzione della posizione del cursore otterremo un valore di tensione su quest'ultimo, dato che funziona come un partitore di tensione.

Dalla scheda Bread Board, con un filo collegheremo la molla di collegamento a 5 V e con un altro filo a quella dello 0 V. Con due fili più lunghi porteremo i due segnali ai capi del potenziometro. Nella figura possiamo vedere il montaggio descritto, dove i 5 V sono collegati



Schema elettrico di un potenziometro.



Muoviamo il potenziometro per cambiare il valore dell'ingresso.

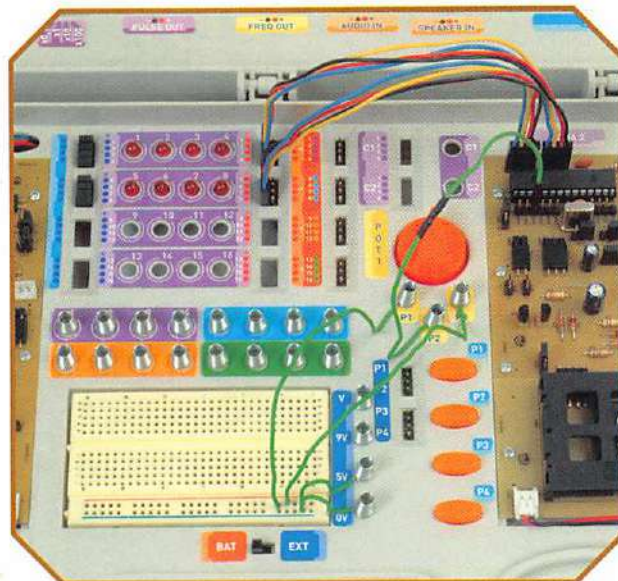
alla molla P3 e 0 V alla molla P1. Dalla molla di collegamento P2, che corrisponde al cursore del potenziometro, otterremo il segnale analogico che dobbiamo collegare all'ingresso.

Attraverso un altro filo di collegamento uniremo P2 con il pin della porta A, configurato come ingresso del convertitore (RA2). L'ingresso sarà correttamente collegato dopo aver terminato il montaggio appena descritto.

## Prova del funzionamento

Per provare il corretto funzionamento del convertitore, è sufficiente ruotare il potenziometro e osservare il risultato sulla matrice dei LED. Con il potenziometro completamente ruotato da una parte, otterremo un ingresso di 5 volt, quindi il risultato della conversione sarà '11111111' il che significa avere tutti i LED accesi. Ruotandolo dall'altra parte applicheremo all'ingresso 0 V, quindi il risultato della conversione sarà '00000000', di conseguenza tutti i LED saranno spenti. Il PIC16F870 dispone di un convertitore a 10 bit, con i quali è possibile rappresentare 1.024 valori ( $2^{10} = 1.024$ ). Se l'ingresso è compreso tra 0 e 5 V, la precisione del convertitore è di  $5/1.024 = 4,88$  mV.

Questo significa che ogni variazione di 4,88 mV sull'ingresso si rifletterà sul risultato della conversione.



Montaggio finale dell'applicazione.

Dato che nel nostro caso stiamo visualizzando solamente gli 8 bit più significativi del risultato della conversione, la precisione sarà di  $5V/2^8 = 5V/256 = 19,5$  mV, quindi ogni 19,5 mV potremo visualizzare una variazione sull'uscita.

Verificheremo inoltre che sono sufficienti piccole variazioni del potenziometro per generare cambiamenti sull'uscita.

## Applicazioni

Immaginate di utilizzare al posto del potenziometro una PT100. La PT100 è una resistenza variabile in funzione della temperatura. Supponete che il range di lavoro di questa resistenza sia fra  $-20^{\circ}$  e  $100^{\circ}$ , quindi il nostro convertitore avrà una precisione di  $120^{\circ}/1.024 = 0,11^{\circ}$ . Rileveremo ogni variazione di  $0,11^{\circ}$ , questo ci permetterà di visualizzare la temperatura con la precisione di un decimo di grado (sul display o su un LCD), controllare il processo in cui è inserita la resistenza con un controllo molto sensibile e preciso.

Immaginate ora di voler costruire invece di un termometro un sensore di pressione digitale o un sensore di umidità. Qualsiasi segnale di tipo analogico si può convertire in uno digitale ed elaborarlo, questo, grazie alle CAD del PIC16F870, si può fare in un modo molto semplice.





# Esercizio 14: la memoria EEPROM, simulazione e pratica

**A**bbiamo sviluppato il codice che risolve l'applicazione dell'esercizio 14 di utilizzo della memoria EEPROM. Abbiamo anche compilato il codice alla ricerca di eventuali errori, e per ottenere il file in codice macchina da scrivere sul PIC. Non abbiamo ancora simulato il funzionamento del programma né realizzato il montaggio. Ora lavoreremo a queste fasi del progetto.

## Simulazione

Dato che per la compilazione abbiamo dovuto creare un progetto associato a questo nostro codice, per simularlo dovremo semplicemente aprire il progetto creato e il file in assembler. Facendo partire MPLAB potremo aprire l'ultimo progetto con cui abbiamo lavorato, mediante una finestra come quella riportata nella figura. Nel caso corrisponda all'esercizio 14 selezioneremo Yes, in tutti gli altri casi selezioneremo No, e continueremo come spiegato precedentemente.

Con il progetto aperto e il codice visualizzato sul monitor, dobbiamo aprire le finestre di simulazione. Sceglieremo la finestra dei Registri delle Funzioni Speciali, la finestra della memoria EEPROM e la finestra dove visualizzare i registri più interessanti per la simulazione (visualizzeremo in binario PORTB, EE\_Dato e EE\_Dir).

Per simulare gli ingressi dovremo utilizzare il simulatore di stimoli asincroni. Nel primo stimolo inseriremo RA4, dato che è l'ingresso che determina la scrittura di un valore nella memoria, continueremo con RA3, RA2, RA1 e RA0. Possiamo porre alcuni degli ingressi

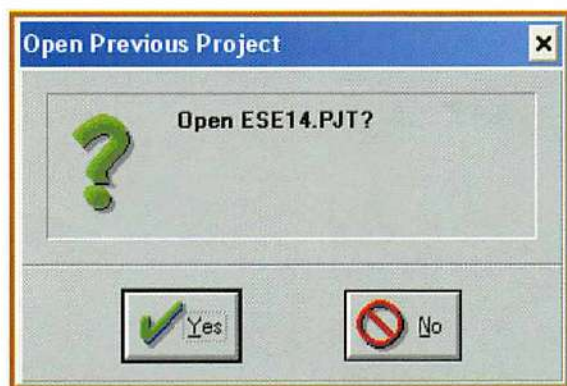
(RA3:RA0) a livello alto (High) prima di iniziare la simulazione passo a passo, ricordate però che per far sì che questi valori siano acquisiti, è necessario cliccare con il mouse su di essi uno alla volta in base a cosa si sta simulando.

Inizieremo la simulazione passo a passo premendo F7. Se la completiamo una volta vedremo che la porta B assumerà il valore '11111111', poiché legge un valore nella memoria. Se prima della volta successiva si clicca su RA4 passandolo a High, vedremo che il programma entrerà nella subroutine di Scrittura, passando alla EEPROM il valore dell'ingresso. Nella subroutine di Lettura si prenderà il valore della EEPROM e lo si porterà sulla porta di uscita.

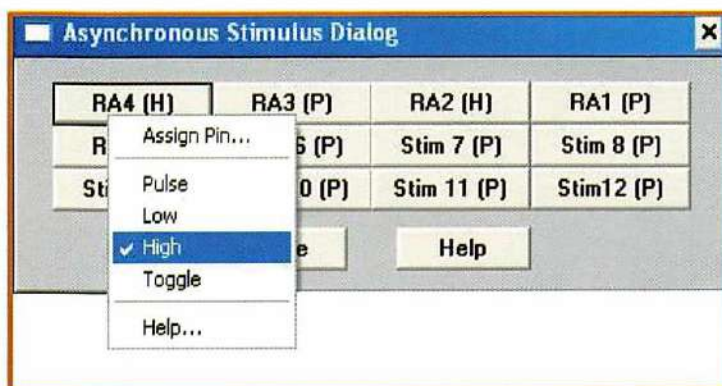
A questo punto la simulazione è avvenuta con successo.

## Scrittura del PIC

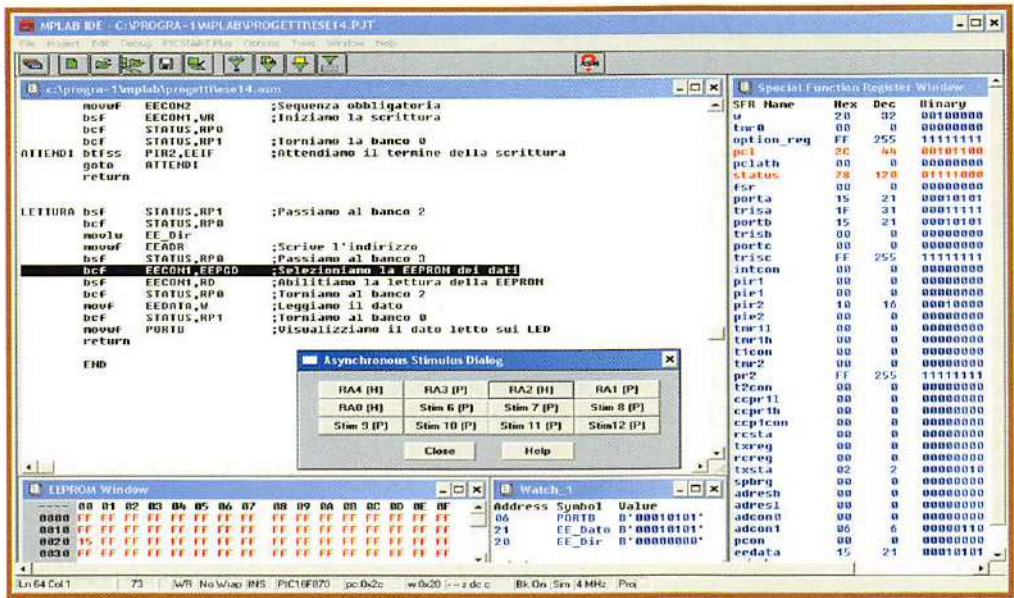
Ricordate che per la scrittura del PIC è necessario predisporre il laboratorio collegando il cavo di trasferimento, inserendo i ponticelli sui connettori JP8 e JP9 e sui connettori JP1, JP2 e JP3 sulle posizioni 1 e 2. Faremo partire



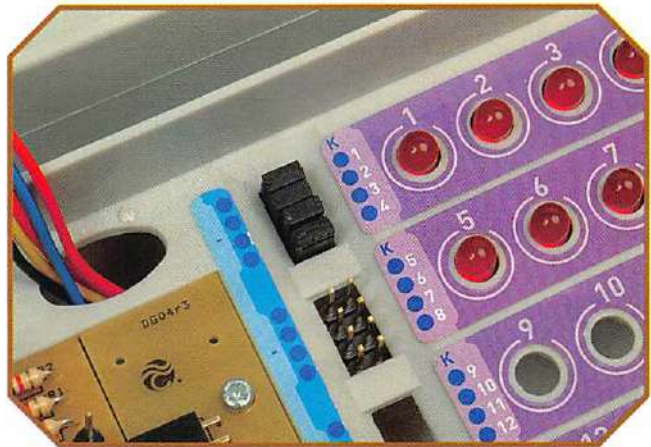
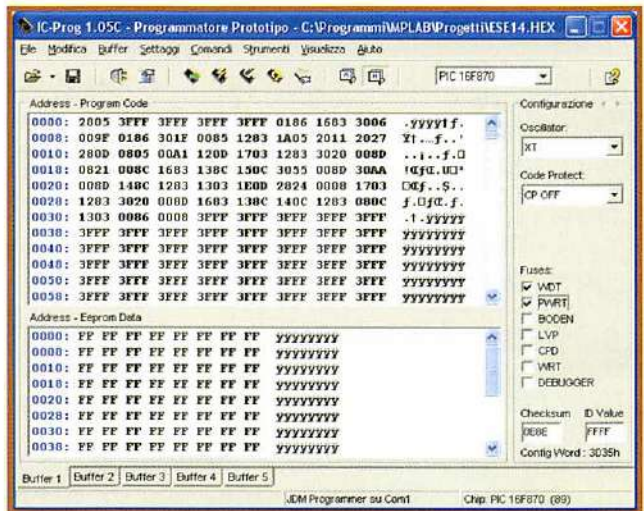
MPLAB ci permette di aprire l'ultimo progetto con cui abbiamo lavorato.



Mediante il simulatore di stimoli asincroni, stimoliamo gli ingressi.

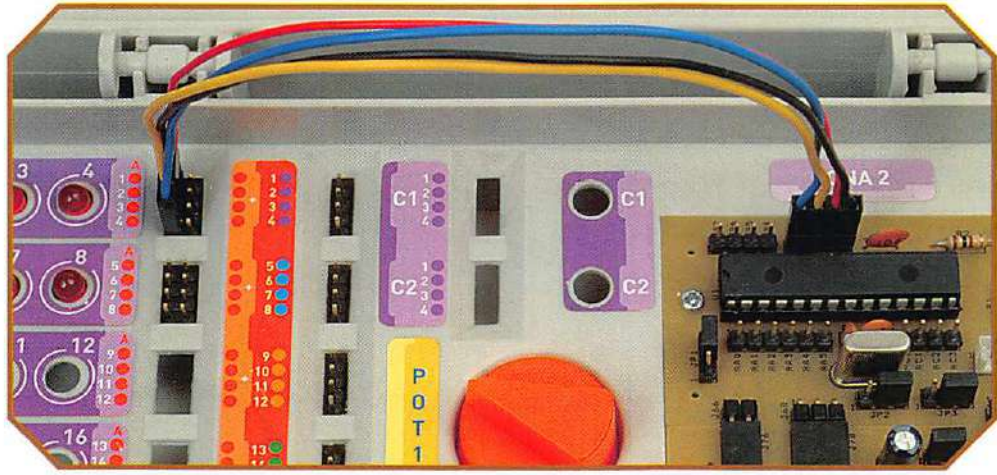


Aspetto di MPLAB durante la simulazione.

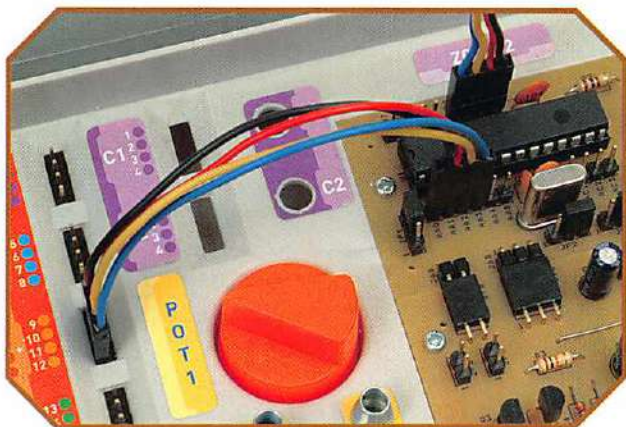


Se utilizziamo la matrice dei LED dobbiamo inserire i ponticelli sui catodi.

Prepariamo IC-Prog per la scrittura del PIC.



Uniamo gli anodi con la porta B.

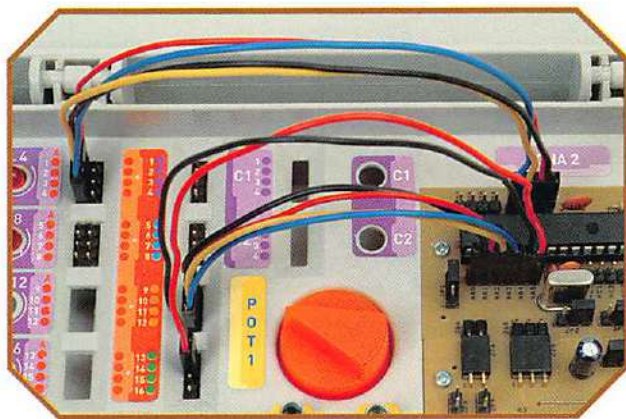


Collegiamo il segnale ai terminali di ingresso.

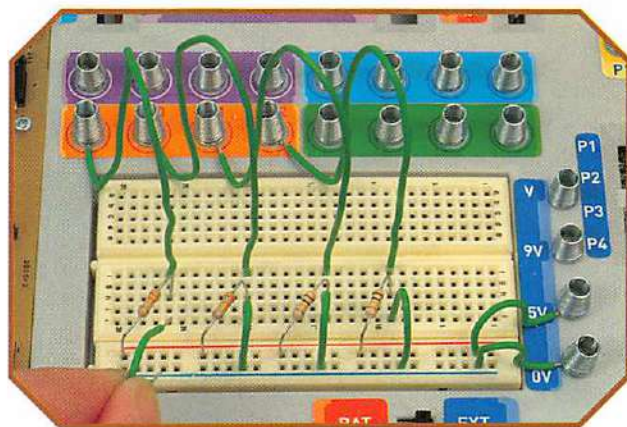
IC-Prog ed eseguiremo le fasi precedenti alla scrittura (lettura, cancellazione e verifica). Apriremo il file che vogliamo trasferire sul PIC e configureremo IC-Prog in modo che il trasferimento si realizzi correttamente, cambieremo il tipo di oscillatore a XT, verificheremo che la protezione del codice sia CP OFF e cliccheremo sui bit WDT e PWRT che fanno parte della parola di configurazione. Fatto questo potremmo continuare con la scrittura selezionando l'opzione Programma Tutto. Terminata la scrittura verificheremo il progetto appena eseguito con una lettura del contenuto del PIC.

## Montaggio delle uscite

Prima di iniziare il montaggio dell'applicazione dobbiamo configurare nuovamente il laboratorio per riportarlo nella condizione pre-



Collegiamo il terminale RA4.



Montaggio dei segnali di ingresso sulla scheda Bread Board.

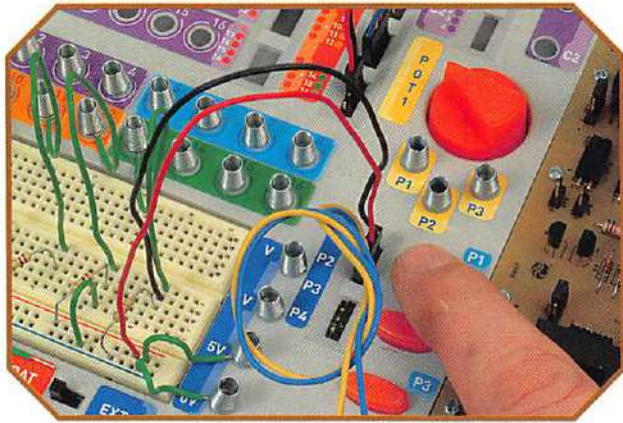
cedente alla scrittura. I ponticelli dei connettori JP8 e JP9 devono essere tolti, quelli dei connettori JP1, JP2 e JP3 devono essere sulle posizioni 2 e 3, ed è necessario togliere il cavo di trasferimento.

Il valore dell'ingresso memorizzato sulla memoria EEPROM esce tramite la porta B visualizzandosi sulla matrice dei diodi LED. Dato che ciò che ci interessa vedere sono i valori corrispondenti agli ingressi RA3:RA0, RA4 lo utilizzeremo solamente per selezionare il momento della scrittura, utilizzeremo solamente i terminali di uscita RB3:RB0. Imposteremo i ponticelli sui catodi della prima fila dei diodi della matrice come si può vedere nella figura, utilizzeremo solamente questa fila di LED, e dato che vogliamo vedere quattro segnali lo faremo mediante un unico cavetto che collegherà gli anodi con i terminali RB3:RB0 della porta B.

## Montaggio degli ingressi

### Ingressi RA3:RA0

Per simulare sul laboratorio gli interruttori collegati a questi ingressi eseguiremo un montaggio sulla scheda Bread Board. I segnali di questa scheda li porteremo alla porta utilizzando le molle di collegamento, quindi uniremo mediante un cavetto i terminale RA3:RA0, con il connettore volante che prende i segnali dalle molle con etichetta arancio (dal 9 al 12). Sulla scheda Bread Board realizzeremo un montaggio molto semplice, mediante delle resistenze. I terminali RA3:RA0,



Tramite il pulsante determineremo quando far passare il segnale di ingresso sull'uscita.

che sarebbe come dire le molle dalla 9 alla 12, si collegano direttamente al positivo tramite una resistenza.

Per cambiare lo stato di questi segnali prepareremo dei ponticelli che potremo collegare al negativo, se vogliamo uno 0 sull'ingresso, o non collegarli se vogliamo un 1. Nella figura possiamo vedere il montaggio che abbiamo appena spiegato.

### Ingresso RA4

Differenziamo questo ingresso dagli altri, perché sarà quello che determina in quale momento i segnali presenti sugli altri terminali di

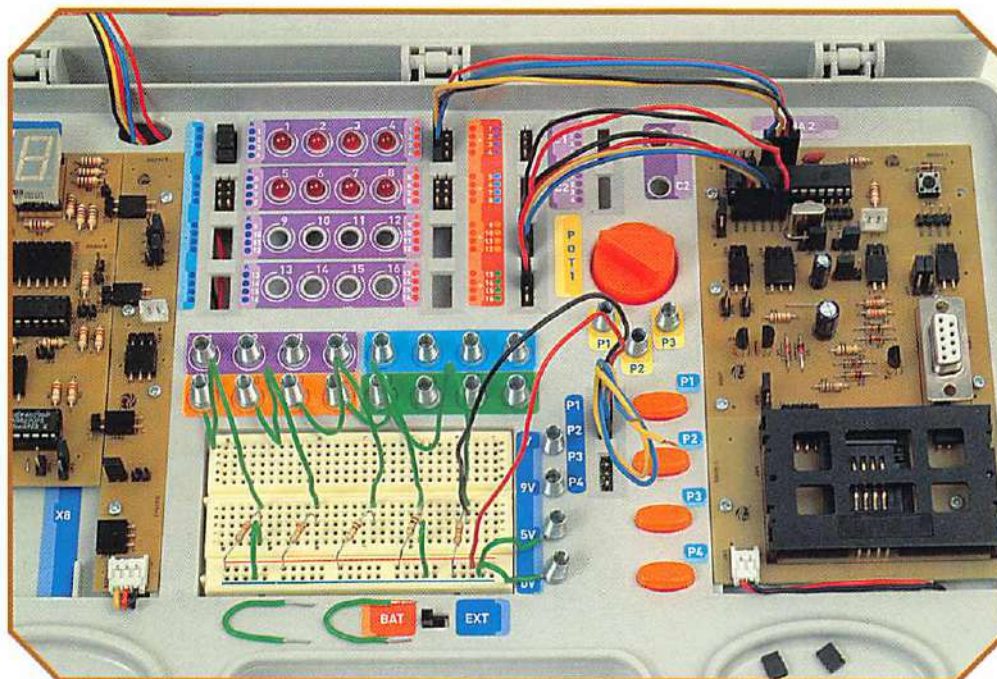
ingresso saranno scritti sulla memoria EEPROM. Per far arrivare questo segnale al terminale corrispondente della porta A (RA4) utilizzeremo le molle di collegamento, quindi potremo unire mediante un cavetto, il connettore che corrisponde al terminale RA4. Nella figura della pagina precedente possiamo vedere questo collegamento.

Alla molla, ovvero allo stesso terminale RA4, collegheremo una resistenza che dall'altro capo andrà al negativo (0V). Invece di utilizzare un ponticello come per i segnali precedenti, potremo attivare il segnale mediante il pulsante P1, che verrà inserito fra la molla e i 5 V. Nella figura si può verificare come viene collegato il pulsante alla scheda Bread Board mediante un cavetto.

### Prova di funzionamento

Per provare il corretto funzionamento impostate alcuni degli ingressi RA3:RA0 al positivo (senza il ponticello) e premete il pulsante P1, vedrete come verrà riportato sull'uscita il valore presente sull'ingresso.

Provate ora a cambiare gli ingressi e verificate che l'uscita non cambia. L'uscita cambia solamente quando si attiva RA4, cioè quando mediante il pulsante passeremo il nuovo valore sulla memoria EEPROM. Se non si attiva P1, il valore visualizzato sarà il valore che il PIC contiene nella memoria EEPROM.



Montaggio dell'applicazione completato.



# Esercizio 16: l'importanza dello SLEEP e del WDT, la pratica

**Q**uesto esercizio è un esempio di come si lavora con l'istruzione **SLEEP** combinata con il **WDT**. È molto importante saper gestire questa istruzione, perché saranno molte le applicazioni che progetteremo in cui il PIC dovrà rimanere in modo riposo o di basso consumo.

## Scrittura del programma

### Configurazione hardware del laboratorio

Per trasferire al PIC il codice in linguaggio macchina (ese16.hex) dobbiamo configurare il laboratorio. Collegheremo il cavo di comunicazione fra quest'ultimo e il PC, verificheremo il corretto posizionamento del PIC sul suo zoccolo e configureremo i ponticelli per questo modo lavoro.

Inseriremo i ponticelli sui connettori JP8 e JP9 e cambieremo di posizione i ponticelli sui connettori JP1, JP2 e JP3 portandoli sulle posizioni 1 e 2.

### Software IC-Prog

Faremo partire IC-Prog e come vediamo dalla tabella nella figura, la prima cosa da fare è leggere il dispositivo nel caso contenesse qualche programma che meriti di essere con-

#### PASSAGGI PER SCRIVERE IL PIC

- 1 Verificare il corretto inserimento del PIC
- 2 Collegare il cavo di comunicazione Laboratorio-PC
- 3 Inserire i ponticelli su JP8 e JP9
- 4 Ponticelli su JP1, JP2 e JP3 sulle posizioni 1 e 2
- 5 IC-Prog: Fase di lettura, cancellazione e verifica
- 6 IC-Prog: Carico del programma e configurazione del software
- 7 IC-Prog: Scrittura e verifica

Passaggi per scrivere un programma nel PIC.

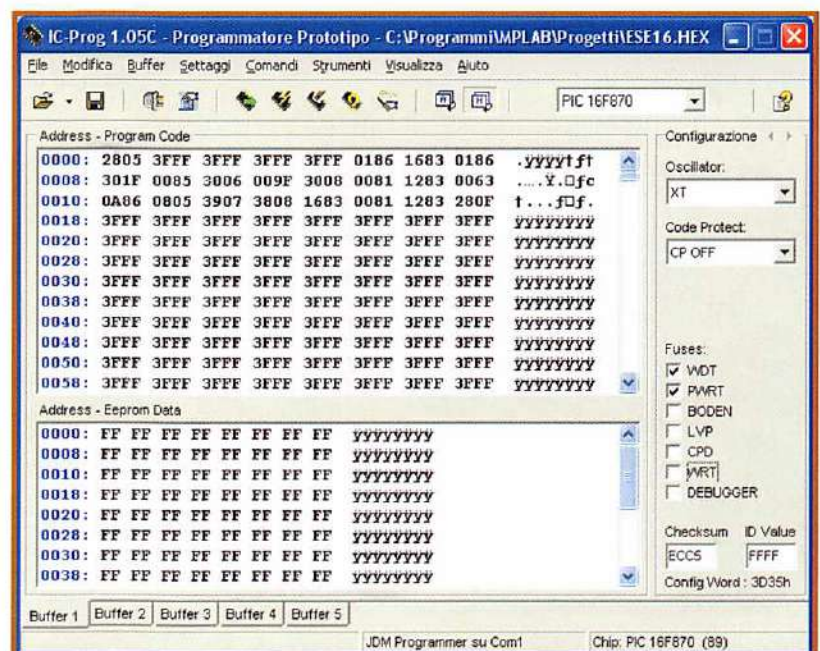
servato. Cancellaremo il contenuto del PIC e verificheremo questa fase per accertarci che sia avvenuta correttamente. Apriremo il nostro file e configureremo l'oscillatore, la protezione del codice e i bit della parola di configurazione.

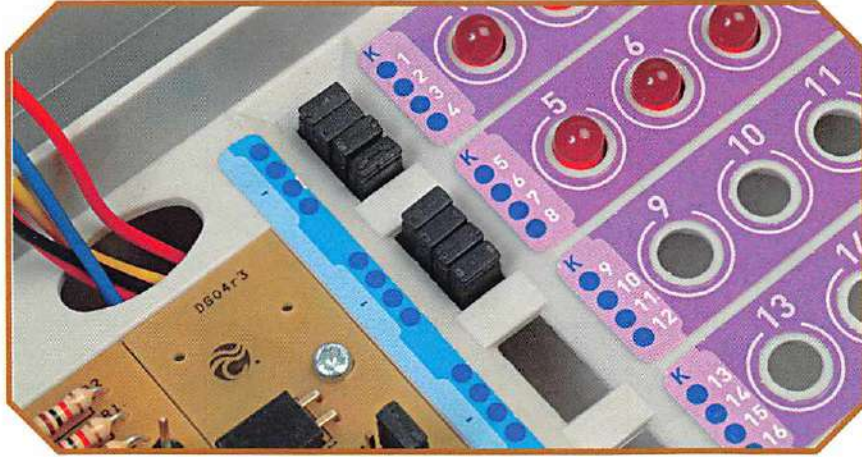
IC-Prog dopo la configurazione avrà l'a-

Porta B		LED n°
RB0	→	4
RB1	→	3
RB2	→	2
RB3	→	1
RB4	→	8
RB5	→	7
RB6	→	6
RB7	→	5

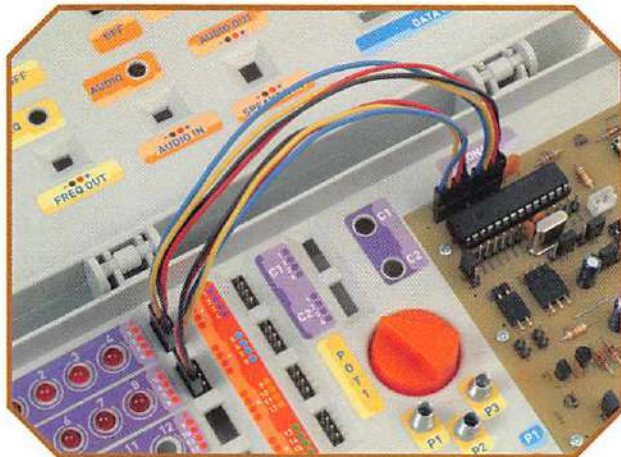
Corrispondenza tra i terminali della porta B e i LED.

Ora possiamo selezionare l'opzione: Programma Tutto.

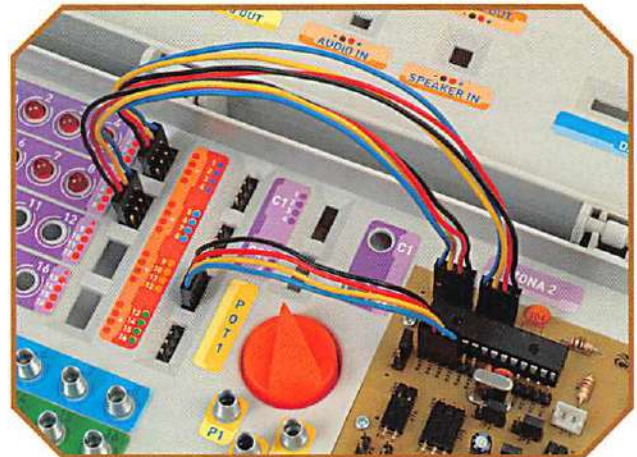




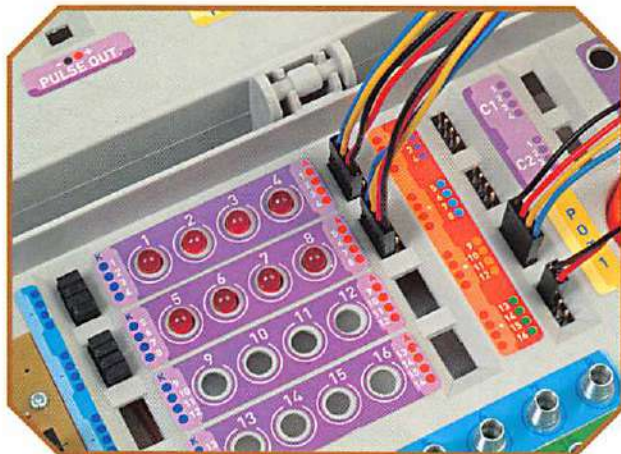
Inseriamo i ponticelli sui catodi dei LED.



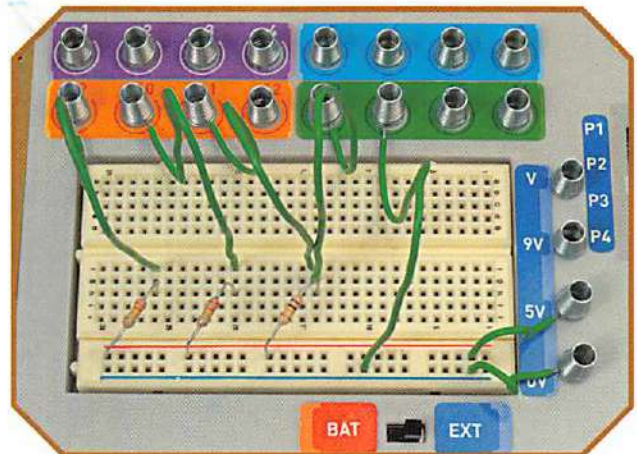
Uniamo con un cavetto la porta B e gli anodi dei LED.



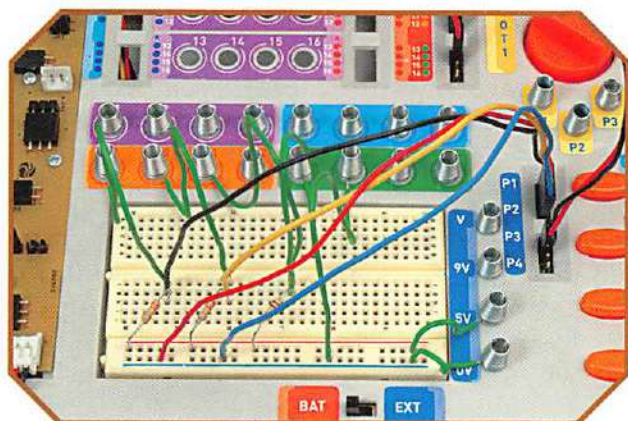
Uniamo la porta A con le molle di collegamento.



Utilizziamo le molle 13 e 14 per inserire il pulsante P3.



Colleghiamo il pulsante P3 sulla scheda Bread Board.



Collegiamo gli altri due pulsanti direttamente sulla scheda.

spetto che possiamo vedere nella figura della prima pagina e sarà pronto per trasferire il programma al PIC. Selezioneremo l'opzione Programma Tutto e inizieremo a trasferire il programma, tenendo presente che al termine del trasferimento dovremo verificare che sia avvenuto con successo. Per eseguire questa verifica è sufficiente leggere il dispositivo e controllare che il programma che si trova scritto all'interno corrisponda a quanto stiamo cercando di scrivere.

## Montaggio

Prima di eseguire il montaggio dell'applicazione dobbiamo riportare la configurazione del laboratorio per il modo lavoro, cioè toglieremo i ponticelli JP8 e JP9 e sposteremo JP1,

JP2 e JP3, inoltre scollegheremo il cavo di trasferimento fra il PC e il laboratorio.

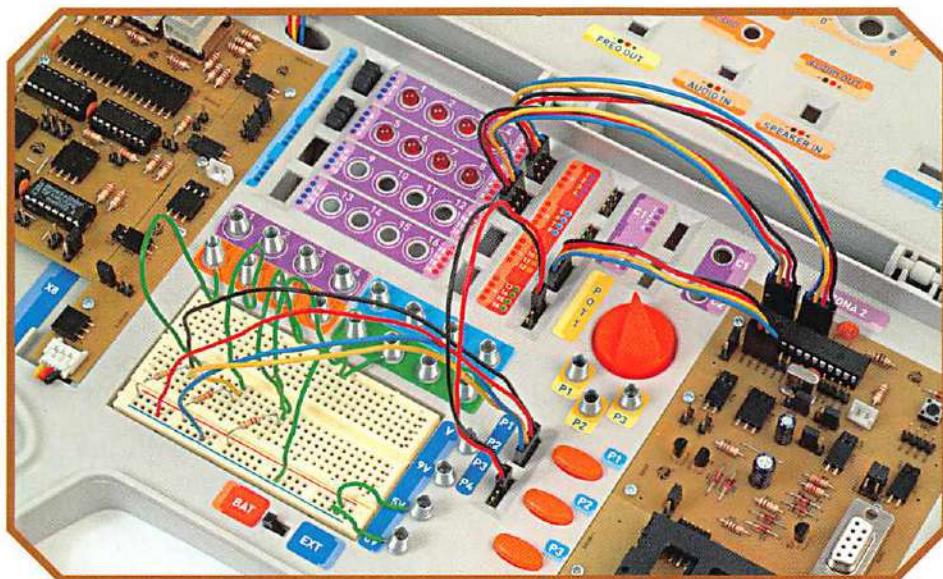
## Montaggio delle uscite

Poiché ogni volta che il PIC si risveglia dal modo lavoro di basso consumo si incrementerà di una unità un contatore e dato che quest'ultimo verrà visualizzato sulla matrice dei diodi LED, dobbiamo collegare la porta B di uscita con la matrice. Prima di eseguire questo collegamento, verificheremo di aver collocato correttamente i ponticelli sui catodi dei diodi. Utilizzeremo due file della matrice, perché il contatore sarà da 8 bit, come si può vedere nella figura.

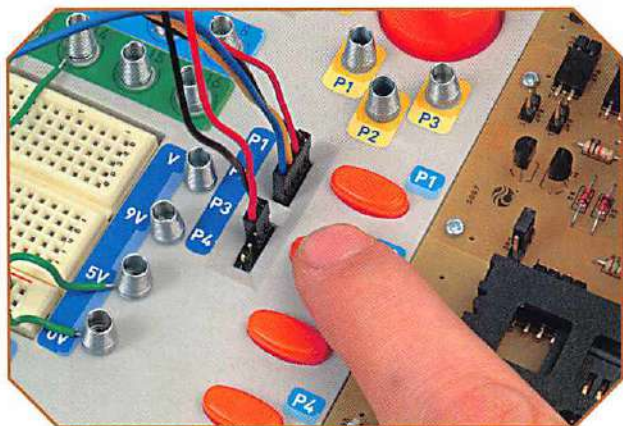
Mediante i cavetti uniremo la porta B con gli anodi dei diodi, tenendo presente che il contatore avrà il bit meno significativo nella fila superiore a destra e il bit più significativo nella fila superiore a sinistra. Nel grafico è possibile vedere la corrispondenza fra i terminali della porta B e i diodi LED della matrice, nelle immagini è possibile vedere i collegamenti.

## Ingressi

Mediante gli ingressi assegneremo il predivisorio del Watchdog Timer, quindi avremo bisogno di tre segnali di ingresso, i quali arriveranno ai terminali RA2, RA1 e RA0. Tramite un cavetto uniremo i terminali della porta A e il connettore maschio corrispondente alla fila



Aspetto del laboratorio dopo aver concluso il montaggio.

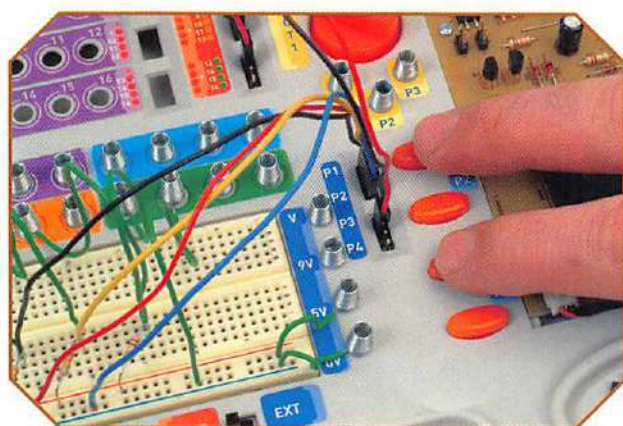


Manteniamo premuto il pulsante per modificare il predivisore.

delle molle di collegamento colore arancio (molle dalla 9 alla 12). Con questo collegamento possiamo quindi eseguire il montaggio degli ingressi sulla scheda Bread Board.

Per simulare gli interruttori utilizzeremo i pulsanti P1, P2 e P3. Negli esercizi precedenti abbiamo visto come ottenere un 1 logico collegando al terminale di ingresso (o dalla sua molla di collegamento equivalente) una resistenza al positivo, per ottenere uno 0 invece uniremo il terminale di ingresso a massa (negativo o 0 V) mediante un cavetto. Per utilizzare i pulsanti sarà sufficiente inserirli fra il terminale di ingresso e la massa, in modo che quando vengono premuti svolgano la stessa funzione del filo che abbiamo utilizzato negli esercizi precedenti.

Monteremo tre resistenze sul positivo nella scheda Bread Board e l'altro capo lo uniremo alle molle di collegamento corrispondenti agli ingressi.



Proviamo diverse combinazioni.

Per collegare il primo dei pulsanti mediante un cavetto aggiuntivo collegheremo i due terminali del connettore maschio relativo al pulsante P3 con il corrispondente connettore delle molle di collegamento 13 e 14 (fila verde).

Una delle molle, la 13 ad esempio, la collegheremo al terminale di ingresso RA2 (molla 11) e l'altra al negativo. In questo modo quando premeremo il pulsante chiuderemo il suo contatto, e porteremo il negativo sull'ingresso del PIC.

Questo è un montaggio valido per collegare i pulsanti, ma come avete potuto verificare abbiamo dovuto utilizzare due molle in più di collegamento. I pulsanti si possono collegare direttamente alla scheda Bread Board senza passare attraverso le molle di collegamento, quindi lo faremo per i due ingressi rimanenti. Collegheremo uno dei capi del pulsante all'ingresso (alla sua molla corrispondente o al collegamento sulla scheda) e l'altro direttamente al negativo. Faremo lo stesso sia per P1 che per P2, in modo che il montaggio risulti simile a quello dell'immagine.

## Prova di funzionamento

Abbiamo terminato il montaggio dell'applicazione e il laboratorio ha un aspetto come quello riportato nell'immagine della pagina precedente. Inizialmente gli ingressi sono tutti al positivo, quindi su RA2, RA1 e RA0 sta entrando '111', ciò significa che il predivisore acquisisce il valore di 128. Questo è il valore massimo che può assumere il predivisore del WDT e significa che la frequenza di incremento del contatore sarà la più bassa, circa 2,3 secondi.

Possiamo cambiare il range del predivisore solamente inserendo uno degli ingressi al negativo. Manterremo P2 premuto, ad esempio, e vedremo che la frequenza del contatore aumenta. Stiamo contando più rapidamente perché ora il segnale dell'ingresso è '101' quindi il predivisore assume un valore di 32 (quattro volte minore di quello iniziale).

Premendo P1 e P3 il range del predivisore è 4 e la frequenza di incremento del contatore è molto più elevata (ogni 72 ms).

Non dimenticate che il microcontroller passa la maggior parte del tempo in modo di basso consumo, uscendo da questo stato solo per incrementare il valore dell'uscita che indica quante volte si è "risvegliato".





# Esercizio 15: la USART, compilazione, simulazione e sviluppo

**N**el fascicolo precedente abbiamo imparato a progettare un programma che stabilisce una comunicazione seriale tra il PC e il PIC tramite il modulo USART di quest'ultimo. Compileremo il programma, lo simuleremo e per i più esperti indicheremo come implementarlo fisicamente e come interagire con il PC.

## Compilazione

Per verificare se il programma preparato non ha errori dobbiamo aprire MPLAB, creare un progetto e nella finestra di edizione di quest'ultimo, aggiungere il file da compilare. Fatto questo selezioneremo Build All per assemblare e compilare il codice.

Il codice compila senza errori e presenta solamente dei messaggi riguardanti la corrispondenza dei registri con i banchi di memoria.

## Simulazione

Per simulare l'esercizio con MPLAB apriremo la finestra dei Registri delle Funzioni Speciali e una finestra con i registri che ci interessa vedere in modo indipendente. Nel nostro caso possono essere: PORTC, aux1, aux2, aux3 e TXREG. La visualizzazione dei dati è consigliabile farla in binario per i primi quattro e in ASCII per l'ultimo. In questo modo potremo vedere più chiaramente se il programma risponde alle nostre aspettative. Nell'immagine presente nella pagina successiva, possiamo vedere come si presenta questa finestra dopo essere stata configurata.

Se cominciamo l'esecuzione del programma

passo a passo, vedremo che questo viene eseguito normalmente fino a quando arriva alla parte di codice che deve inviare i caratteri ed entra nella subroutine di invio. La simulazione si fermerà in un ciclo di attesa fino a quando non terminerà la trasmissione. Per poter uscire da questo ciclo, forzeremo un valore sul registro TXSTA utilizzando la finestra Modify. Nell'immagine in figura è riportato un esempio di come forzare un valore all'interno del registro.

Se forziamo il valore del registro TXSTA a FF (in binario '11111111'), ogni volta che entra nella subroutine di trasmissione uscirà da questa senza problemi.

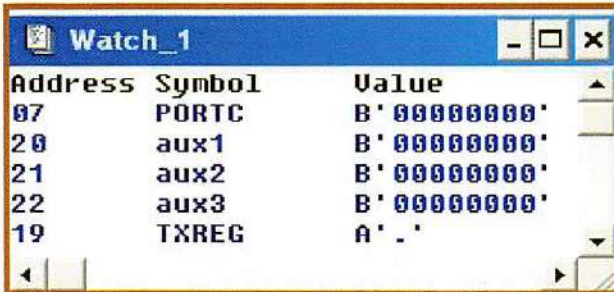
Continuando a eseguire il programma vedremo che, dopo aver letto il valore della Porta A, viene invertito l'ordine dei bit, di conseguenza il valore della variabile aux3 cambia, fino a ottenere l'inversione completa del valore letto sulla porta; successivamente si trasmette il valore ASCII ottenuto dalla combinazione dei pin di ingresso. Se non simuliamo nessun valore di ingresso verranno trasmessi cinque '0' e non potremo vedere come viene riordinato il valore da trasmettere, ma se nella esecuzione successiva, prima di acquisire il valore della porta, simuleremo degli stimoli sui pin, potremo vedere tutti i passi eseguiti in modo corretto.

```
Build Results
Building ESE15.HEX...

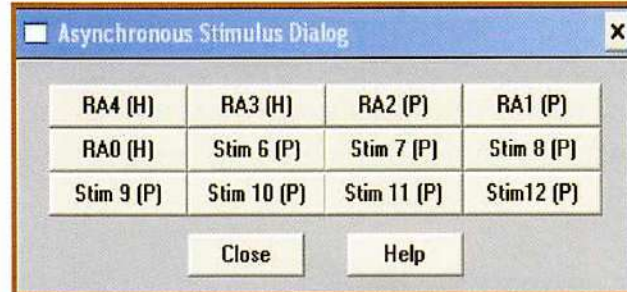
Compiling ESE15.ASM:
Command line: "D:\PROGRA~1\MPLAB\MPSASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 31 : Register in operand not in bank 0. Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 42 : Register in operand not in bank 0. Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 44 : Register in operand not in bank 0. Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 46 : Register in operand not in bank 0. Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 48 : Register in operand not in bank 0. Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 50 : Register in operand not in bank 0. Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\ESE15.ASM 53 : Register in operand not in bank 0. Ensure

Build completed successfully.
```

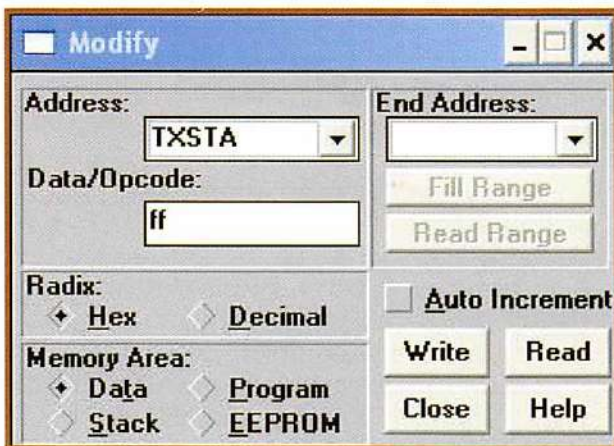
Risultato della compilazione.



Finestra dove si vede il valore dei registri più interessanti.



Con il simulatore di stimoli asincroni forziamo il valore di ingresso.

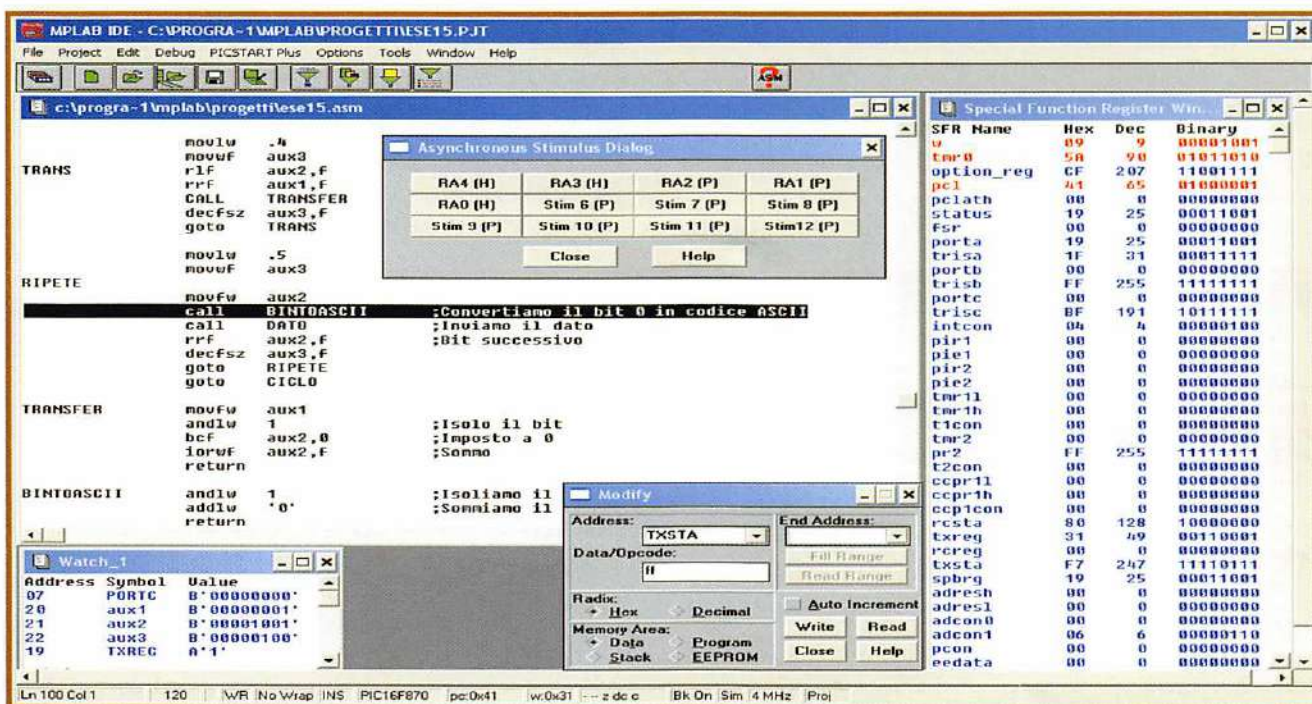


Forziamo il valore del registro TXSTA per uscire dal ciclo.

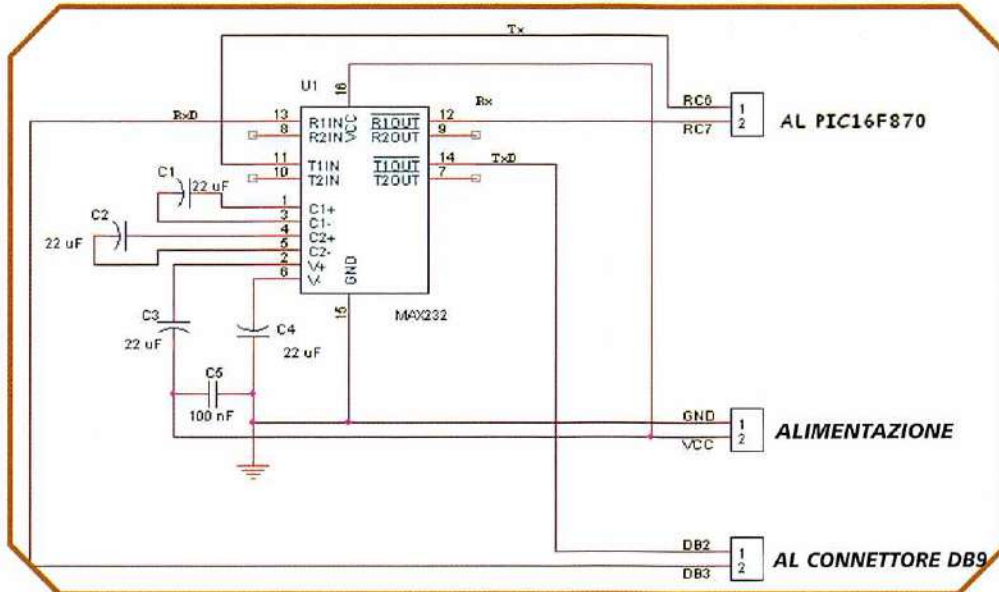
Con il simulatore di stimoli asincroni simuliamo un ingresso qualsiasi, ad esempio quello riportato nella figura. Ricordate che per attivare il livello alto sui pin selezionati, dovremo cliccare con il pulsante sinistro del mouse sul pulsante corrispondente al pin, uno per uno. Attiveremo un pin, eseguiremo una linea di programma, attiveremo un altro pin eseguiremo un'altra linea, e così via. Verificheremo che in queste condizioni, al momento della simulazione, il programma risponde correttamente.

### Implementazione e sviluppo

Vediamo ora come si implementa questo esercizio fisicamente, per quei lettori che vorran-



Aspetto di MPLAB durante la simulazione.



Schema elettrico della comunicazione via seriale RS232.

UNITÀ	COMPONENTE
1	Circuito integrato RS232 (p.e. MAX232)
4	Condensatori da 22 $\mu$ F
1	Condensatore da 100 nF
1	Connettore DB9 femmina

Componenti necessari per il montaggio del circuito elettrico di comunicazione RS232.

no "andare più avanti", e desiderano svilupparlo correttamente.

Per stabilire una comunicazione seriale con il modulo USART è necessario disporre di un circuito esterno al microcontroller che implementi il protocollo di comunicazione. Il protocollo che si utilizza in questi casi è il RS232.

## Circuito di ingresso

Dobbiamo usare cinque interruttori per gli ingressi. Nel caso non avessimo a disposizione gli interruttori necessari potremo eseguire un semplice montaggio basato sulle resistenze, come abbiamo visto negli esercizi precedenti.

## Montaggio necessario per stabilire la comunicazione

Dobbiamo comunicare usando il protocollo RS232, quindi è necessario eseguire un montaggio specifico per questa comunicazione. Un circuito elettrico tipico per comunicazioni

RS232 è quello riportato nella figura. Come potete verificare, abbiamo specificato nel circuito con quali terminali del PIC si collega (RC6 e RC7), come si collega l'alimentazione e a quali pin del connettore DB9 deve essere unita la linea di comunicazione.

Tramite il terminale RC6 del PIC uscirà il dato da trasmettere, che entrerà nel circuito integrato tramite il terminale 11 (T1IN). Il dato in formato RS232, uscirà dall'integrato tramite il terminale 14 (T1OUT) verso il connettore DB9, sul suo terminale DB2. Quando il PC invia un dato, questo arriva sul circuito tramite il pin DB3 del connettore e arriva al circuito integrato tramite il terminale 13 (R1IN). L'integrato converte il dato per poterlo trasmettere al PIC, facendolo uscire tramite il pin 12 (R1OUT) per arrivare a questo tramite il terminale RC7.

Avendo chiari questi collegamenti, è sufficiente inserire i condensatori indicati nello schema, alimentare il circuito integrato e unire il terminale 5 (DB5) del connettore DB9 a massa.

## Hyper Terminal di Windows

Parliamo ora di questo programma, anche se non è l'unico software di comunicazione con il quale può funzionare la nostra applicazione. Questo programma si trova nel menù Comunicazioni, all'interno di Accessori, nel menù a tendina di Programmi. Facendo il dop-



Apriamo Hyper Terminal e creiamo una connessione.

pio clic sull'icona del programma apparirà una videata per creare una nuova comunicazione. In questa finestra dobbiamo dare un nome al collegamento che vogliamo creare e possiamo scegliere l'icona che ci risulta più appropriata. Fatto questo, la videata successiva serve per stabilire il dispositivo tramite il quale si stabilirà la comunicazione: modem, sockets o porta. Nel nostro caso sceglieremo



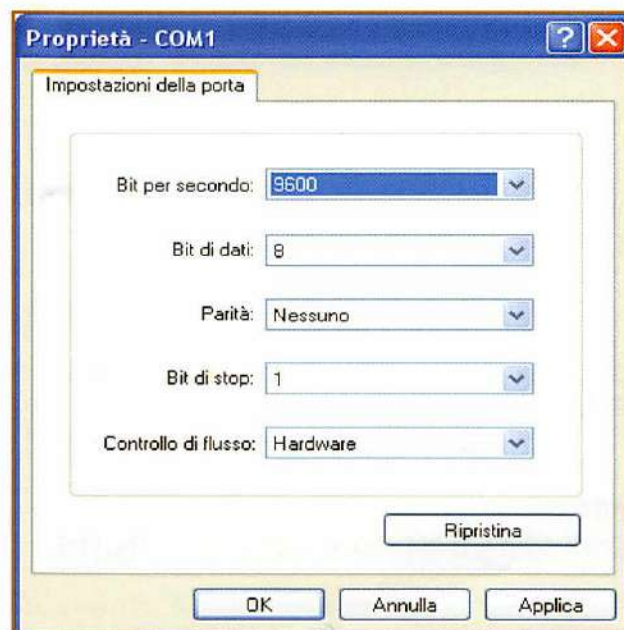
Dispositivo tramite il quale si effettua la comunicazione.

la porta seriale COM1, dove collegheremo il nostro cavo di comunicazione. Dopo aver scelto il dispositivo dovremo configurarlo, e questo lo si fa mediante una nuova finestra. Nella finestra Proprietà di COM1, dobbiamo configurare i parametri della nostra comunicazione, che nel nostro caso saranno 9.600 bit per secondo, 8 bit di dati, parità pari, un bit di stop e controllo di flusso di tipo hardware. Potete provare a cambiare alcuni di questi parametri, osservando che cosa succede nella comunicazione. Configurata la comunicazione si apre una finestra di lavoro in cui verranno visualizzati i messaggi inviati tramite il PIC.

## Avvertenze

I componenti per montare il circuito di comunicazione seriale 232 non sono forniti con l'opera, quindi chi fosse interessato alla realizzazione di questo montaggio, li dovrà acquistare in un qualsiasi negozio di componenti elettronici.

In ogni caso l'obiettivo principale era l'acquisizione delle conoscenze necessarie per lavorare con qualsiasi dei dispositivi del PIC, quindi per quanto riguarda la USART consideriamo raggiunto il nostro obiettivo.



Configurazione dei parametri della comunicazione.



# Esercizio: controllo del forno; compilazione, simulazione e sviluppo

**C**ontinuiamo il progetto del forno di modellazione allo scopo di completare tutte le fasi del progetto e ottenere un risultato ottimale. Compileremo il codice, lo simuleremo e realizzeremo un montaggio parziale del circuito elettrico, lasciando ai più esperti il montaggio completo dello stesso.

## Compilazione

Il codice che svilupperemo per risolvere l'applicazione non è ancora stato messo a punto. Dobbiamo quindi, compilare questo codice e correggere gli eventuali errori che potrebbero essere contenuti in esso. È normale commettere errori e trovarli in fase di compilazione, tanto più se il codice è lungo e presenta una certa difficoltà. L'esempio che potete trovare sul secondo CD-R "Forno 75-125.asm" verrà utilizzato per seguire tutte le fasi del progetto (compilazione, simulazione e montaggio), tenendo presente che per il codice "Forno 100-150.asm", è praticamente la stessa cosa, a parte il risultato della simulazione, che cambierà in base ai valori limite della temperatura.

In MPLAB creeremo un progetto a cui allegheremo il codice da compilare e simulare. Sceglieremo l'opzione Build All ed eseguiremo la compilazione e l'assemblamento del codice. Il codice dell'esempio è compilato con successo, ma presenta alcuni messaggi nuovi,

compresi tre avvisi di "warning" di possibili errori. Gli ultimi quattro messaggi ci informano che non avendo specificato la destinazione del risultato dell'operazione, questa prenderà il proprio registro come destinazione. Le tre warning si producono quando inseriamo il valore 325000 per ottenere il tempo di modellazione, dato che il compilatore stima questo valore fuori dal range. In altre parole non è compreso fra i valori accettati dal registro di lavoro.

È necessario prestare sempre particolare attenzione agli avvisi di questo tipo (warning), dato che possono provocare un errato funzionamento del programma, nel nostro caso però continueremo con lo sviluppo attendendo il montaggio finale per verificare se l'errore esiste realmente.

## Simulazione

Dobbiamo simulare tutto ciò che è possibile per poter passare alla fase del montaggio con la sicurezza che il programma che girerà sul

```
Build Results
Building FORNO7~1.HEX...

Compiling FORNO7~1.ASM:
Command line: "D:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 46 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 47 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 49 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 51 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 53 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 55 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 57 : Register in operand not in bank 0. Ensure that
Warning[202] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 60 : Argument out of range. Least significant bits u
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 67 : Register in operand not in bank 0. Ensure that
Warning[202] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 87 : Argument out of range. Least significant bits u
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 120 : Register in operand not in bank 0. Ensure that
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 122 : Register in operand not in bank 0. Ensure that
Warning[202] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 142 : Argument out of range. Least significant bits
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 154 : Register in operand not in bank 0. Ensure that
Message[305] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 237 : Using default destination of 1 (File).
Message[305] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 239 : Using default destination of 1 (File).
Message[305] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 254 : Using default destination of 1 (File).
Message[305] C:\PROGRA~1\MPLAB\PROGETTI\FORNO7~1.ASM 256 : Using default destination of 1 (File).

Build completed successfully.
```

Risultato della compilazione.



### Messaggio:

```
incf    aux21    incf    aux31
subwf   aux11    subwf   aux11
```

Utilizzano come registro di destinazione lo stesso registro su cui avviene l'operazione.

Messaggio che otterremo durante la compilazione.

### Warning:

```
imovlw  .325000
```

L'argomento è fuori range.  
 Il compilatore fa riferimento a un registro di lavoro da 8 bit, quindi accetta valori da 0 a 255 ( $2^8 = 256$ ).  
 È possibile inserire un numero così grande?  
 Lo verificheremo montando l'esercizio pratico e studiandone il risultato.

Avviso di pericolo (Warning) che riguarda tre delle linee del codice.

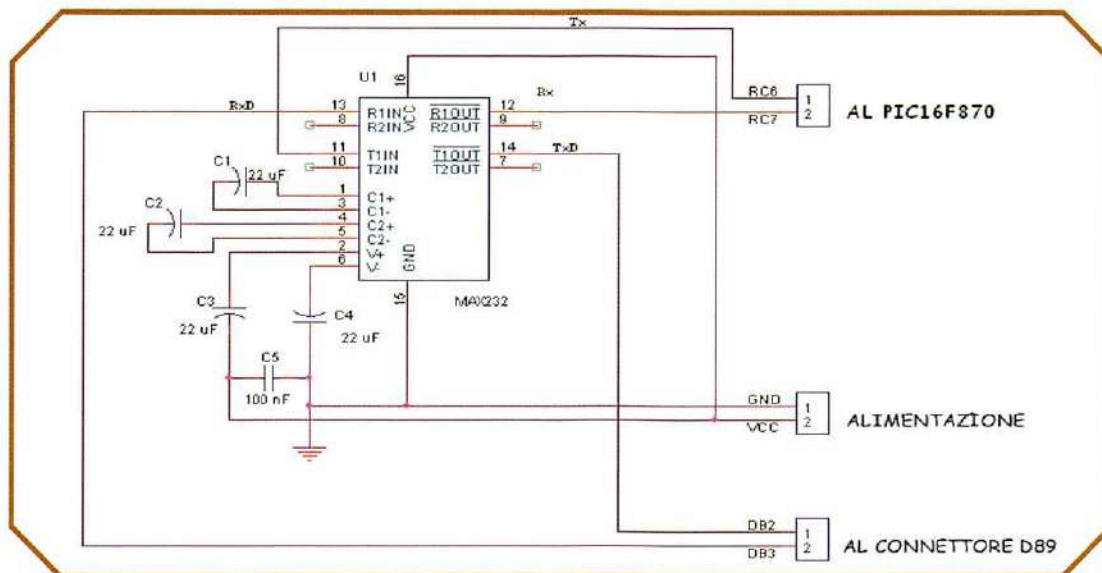
Address	Symbol	Value
06	PORTB	B'00000000'
19	TXREG	A'█'
1E	ADRESH	B'00000000'
27	cont	D'136'
28	cont1	D'136'

Finestra dove possiamo vedere i registri più importanti.

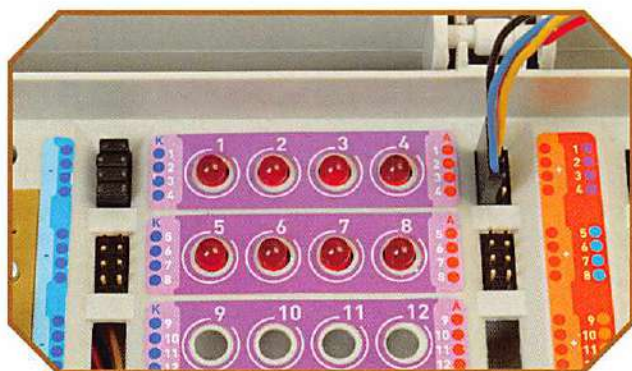
PIC risponda in modo soddisfacente. Apriamo la finestra dei Registri delle Funzioni Speciali e una finestra dove visualizzare i registri più importanti (PORTB, TXREG, ecc.). Ricordatevi che per visualizzare questi registri sarà opportuno impostarli nel formato più adeguato (PORTB in binario, TXREG in ASCII...). Se iniziamo a simulare, il programma percorrerà correttamente il codice fino a quando dovrà inviare i dati tramite la USART, momento in cui il programma stesso si ferma e attende che il dato venga trasmesso. Per uscire da questa attesa è necessario ricordare il trucco che ab-

The screenshot shows the MPLAB IDE environment. The main window displays assembly code with comments in Italian. A 'Special Function Register Window' is open on the right, listing various registers like PORTB, TXREG, and ADRESH. A 'Watch\_1' window is also open, showing the values of these registers. A 'Modify' dialog box is visible in the foreground, allowing the user to change the address and data/opcode of a selected instruction.

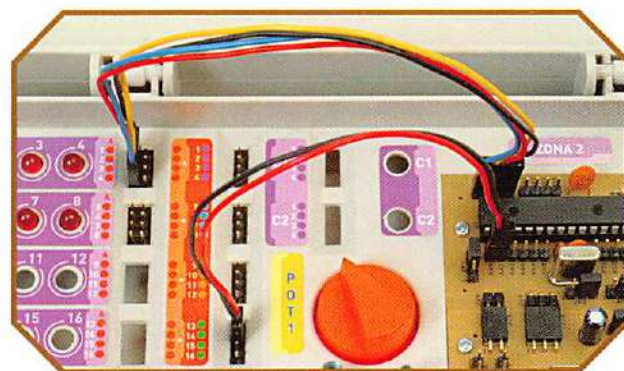
Videata di MPLAB durante la simulazione.



Circuito di comunicazione.



Collegiamo le uscite alla matrice dei LED.



Uniamo gli ingressi alle molle di collegamento.

biamo utilizzato nell' Esercizio 15, che consisteva nel forzare un valore sul registro TXSTA utilizzando la finestra Modify. Se forziamo il valore del registro TXSTA a FF (in binario '1111111') ogni volta che entra nella subroutine di trasmissione uscirà da essa senza problemi. Possiamo vedere come evolve il programma se continuiamo con la simulazione.

### Caricamento del programma sul PIC

Fate partire IC-Prog ed eseguite i passaggi necessari per caricare il programma sul PIC.

Ricordatevi che è necessario configurare il laboratorio tramite i jumpers per stabilire la comunicazione con il PC e poi tornare a riconfigurarlo per provare il funzionamento del progetto. Cancellate il contenuto, aprite il file in codice macchina (Forno 75-125.hex), configurate l'oscillatore e i bit della parola di con-

figurazione e trasferite il programma sul PIC.

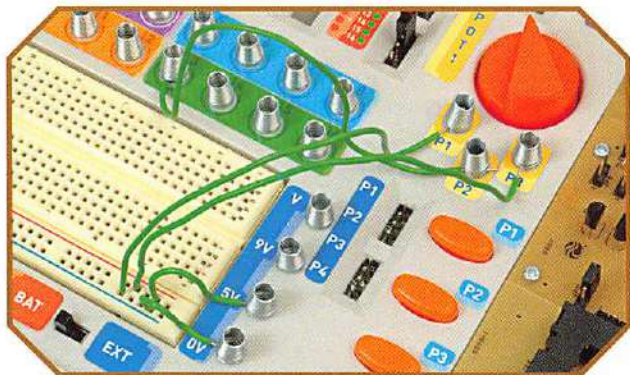
### Montaggio e sviluppo

Come già succedeva per l'esercizio 15, per eseguire il montaggio completo di questo progetto è necessario avere i componenti aggiuntivi per stabilire la comunicazione fra il PIC e il PC. Il circuito di comunicazione è basato sullo schema elettrico riportato nella figura in alto, dove tramite un circuito integrato 232 è possibile stabilire il protocollo di comunicazione.

Ciò che dobbiamo realizzare è il montaggio elettrico del resto del circuito, dato che lavorando con dispositivi quali il CAD risulta essere molto interessante.

### Montaggio delle uscite

Eseguiamo un montaggio per verificare che le uscite RB4 e RB5, indicatori di allarme, funzio-

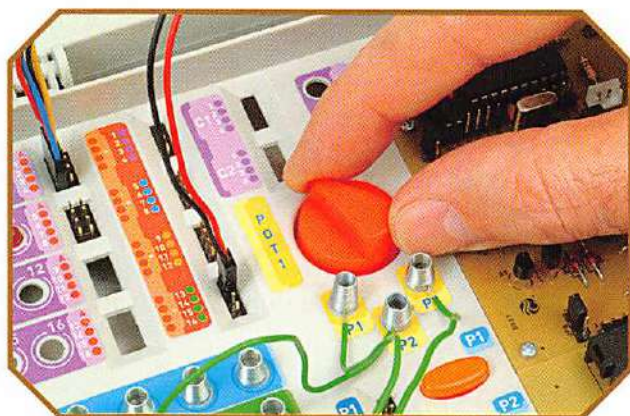


Collegiamo il potenziometro.

nino in modo adeguato. A questo scopo inserite i ponticelli sui catodi della matrice dei LED – dato che proveremo gli allarmi tramite questi diodi – e unite mediante un cavetto le uscite del PIC con la matrice dei LED.

## Montaggio degli ingressi

Il segnale di ingresso per il nostro progetto, dovrebbe essere il segnale di una sonda di temperatura collocata all'interno del forno di modellazione e come abbiamo detto questo ingresso è possibile simularlo con un potenziometro. Collegheremo tramite un cavetto i terminali di ingresso del PIC con il connettore diritto maschio corrispondente alle molle di collegamento della fila di colore verde. Utilizzeremo il potenziometro del laboratorio e per questo lo dovremo collegare all'alimentazione. Mediante i fili di collegamento uniremo i capi di quest'ultimo a 5 e a 0 V rispettivamente, essendo il cursore o elemento mobile la parte che deve essere collegata all'ingresso.



Muoviamo il potenziometro per osservare la risposta del circuito.

All'ingresso del PIC otterremo quindi un segnale analogico compreso fra 0 e 5 V. Nell'immagine della figura è riportato un esempio di come poter realizzare questo collegamento.

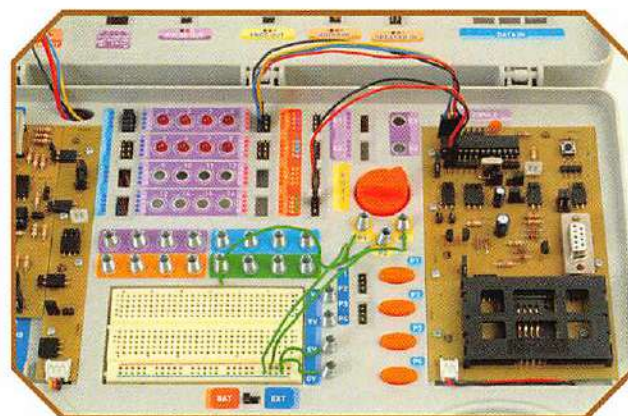
## Prova del circuito

Per provare il corretto funzionamento del montaggio eseguito dobbiamo "giocare" con il potenziometro, in modo che partendo dalla sua posizione estrema, in cui i due diodi sono spenti, muoveremo poco a poco fino a ottenere il primo allarme in cui uno dei diodi inizia a lampeggiare. In questo momento simuleremo il raggiungimento del primo livello di temperatura (75°). Se continuiamo a ruotare il potenziometro arriveremo al punto in cui il LED che lampeggiava resta acceso in modo fisso, e l'altro LED che indica l'ingresso di un nuovo pezzo nel forno si accende passato il tempo prestabilito.

## Conclusioni

Il progetto completo comprende anche il programma che prevede l'altro tipo di pezzi (Forno 100-150.asm). Questo è identico, tranne per i limiti di temperatura, in tutte le fasi inclusa quella del montaggio, a quanto abbiamo appena esposto.

Quando inizieremo l'utilizzo del programma residente Bootloader o Uploader diventerà più semplice caricare un nuovo programma sul PIC, senza dover cambiare la configurazione hardware, quindi nei casi come questo, dove è necessario avere un programma per ogni tipo di pezzo, risulterà molto comodo.



Aspetto del laboratorio con il montaggio sopra descritto.





# Allarme acustico con memoria

**Q**uesto esperimento consiste in un circuito che si attiva quando si apre un circuito elettrico, segnalando in modo visivo e acustico, e mantenendo bloccati entrambi i segnali, anche se il circuito torna a chiudersi, l'allarme si disattiva con un pulsante, dopo aver stabilito il collegamento precedentemente aperto.

## Il circuito

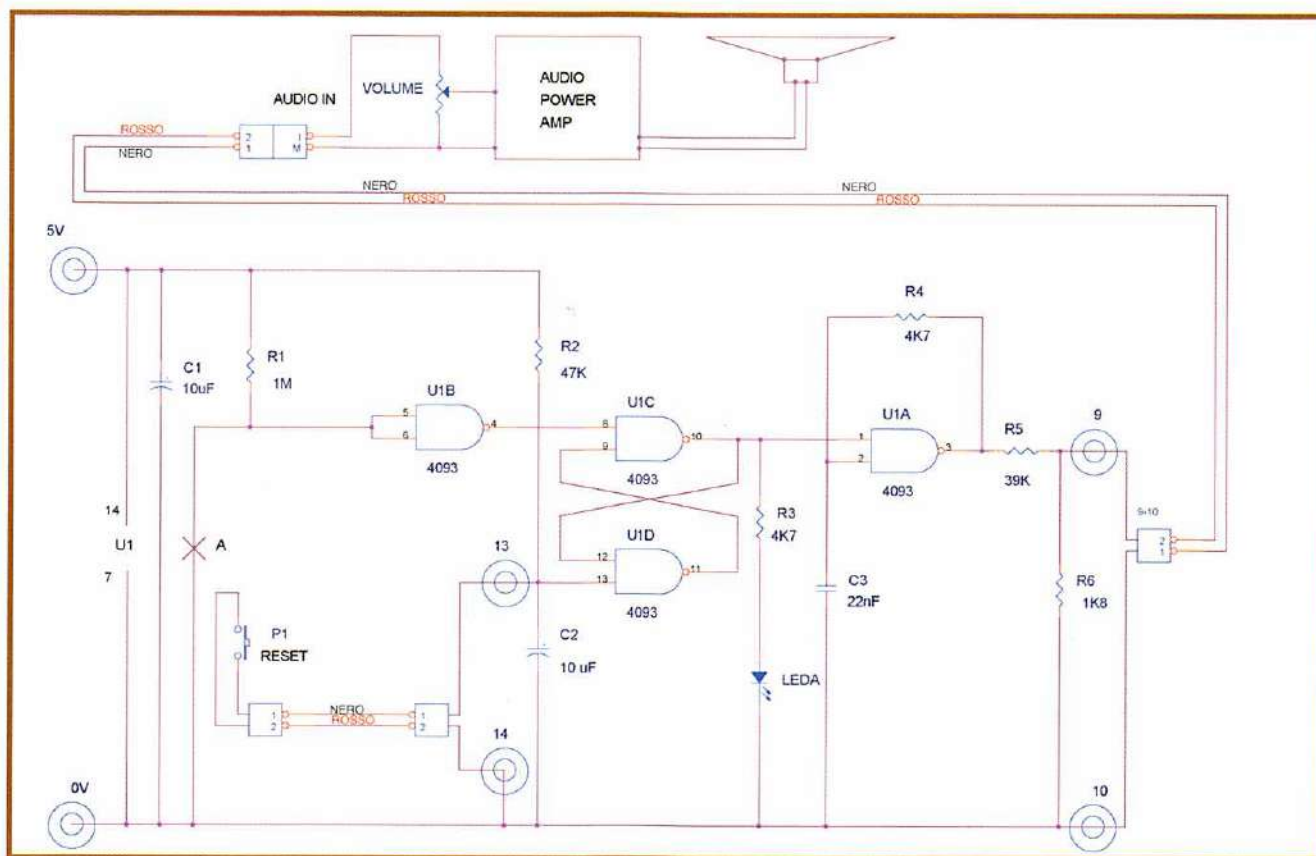
Questo tipo di circuito si può utilizzare per indicare stati di allarme, ad esempio per rilevare l'apertura di una porta o l'apertura di un filo.

Il circuito è piuttosto semplice, inoltre parte di esso ci è già nota.

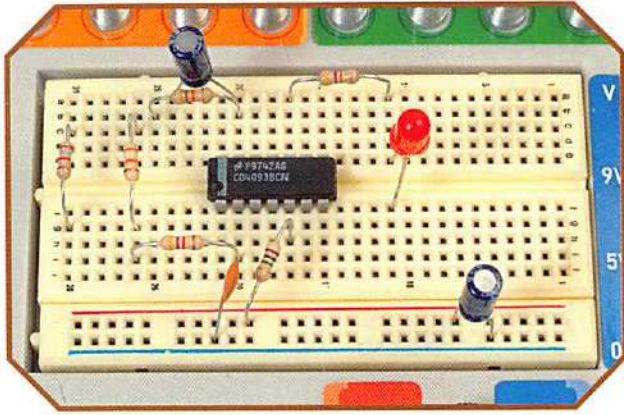
L'elemento che memorizza la rottura del filo è un bistabile RS, formato dalle porte U1C e U1D del circuito integrato 4093; ricordiamo

che gli ingressi di questo circuito bistabile sono attivi a livello basso. L'ingresso di SET è il terminale 8 del 4093 e l'ingresso di RESET è il terminale 13 dello stesso integrato.

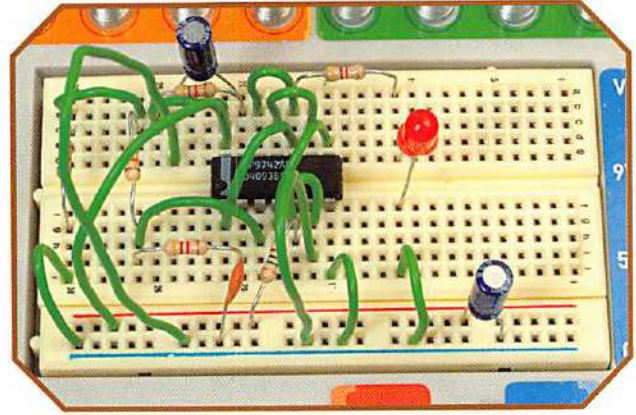
La porta U1B si utilizza per rilevare lo stato dell'ingresso ed è configurata come invertente, avendo i suoi due ingressi collegati insieme. Nella posizione di riposo l'ingresso di questa porta invertente è 0, dato che è collegata al negativo dell'alimentazione mediante un filo. Quando questo filo si scollega, o si taglia,



Schema elettrico del circuito di allarme acustico con memoria.



Componenti sulla scheda Bread Board.



Cablaggio interno della scheda.

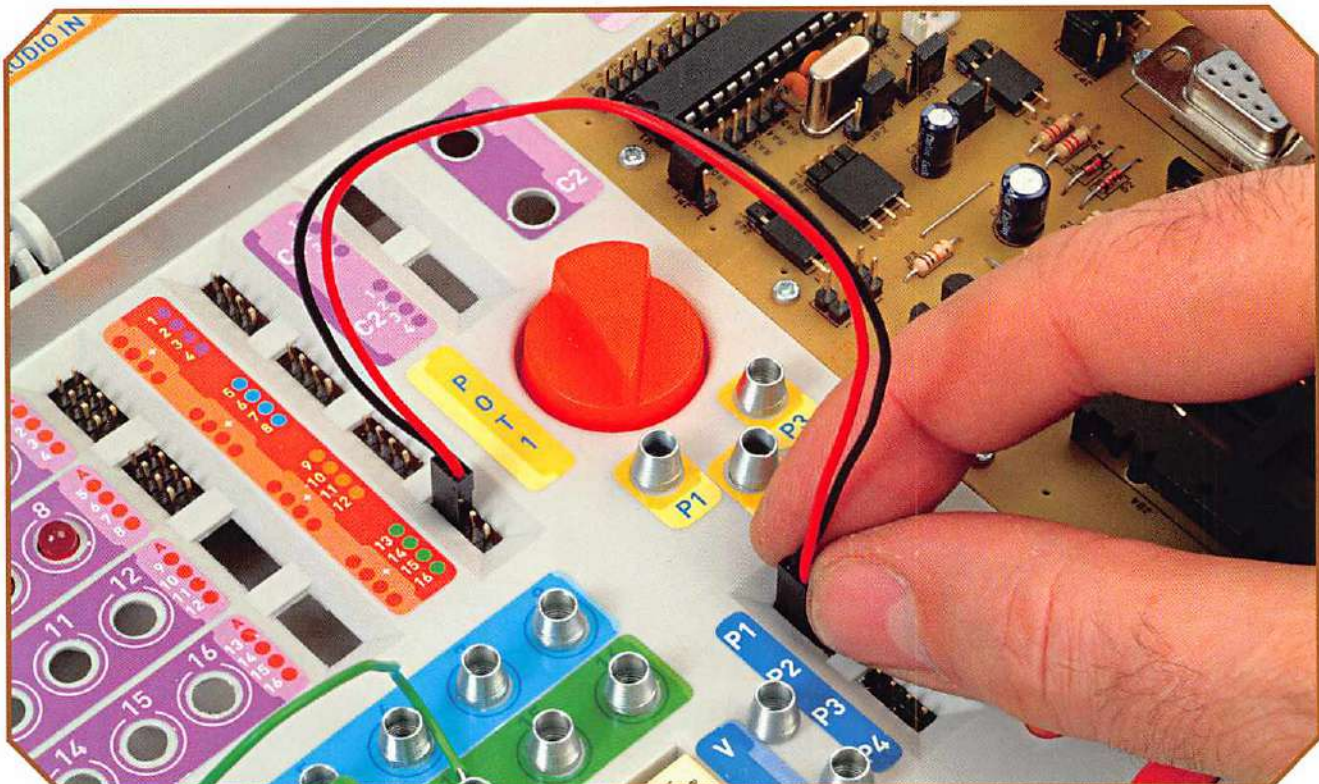
la resistenza R1 forza un livello alto sull'ingresso della porta e la sua uscita passa a livello basso, il terminale 10 a livello alto, illuminando il LED A.

Il terminale 10 è anche collegato all'ingresso della porta U1A che a sua volta è collegata in modo tale da funzionare come un oscillatore astabile, fornendo sull'uscita un segnale la cui frequenza fondamentale rientra all'interno della banda udibile. Quando il terminale 1

del 4093 è a livello basso, non si genera segnale audio. La frequenza di questo oscillatore dipende dai valori della resistenza R4 e dal condensatore C3.

L'uscita di questo oscillatore si attenua tramite il partitore resistivo formato dalle resistenze R5 e R6, prima di essere applicata all'ingresso dell'amplificatore audio.

Aprendo il circuito nella zona indicata nello schema con una croce e la lettera A, questo si



Collegamenti del pulsante P1.

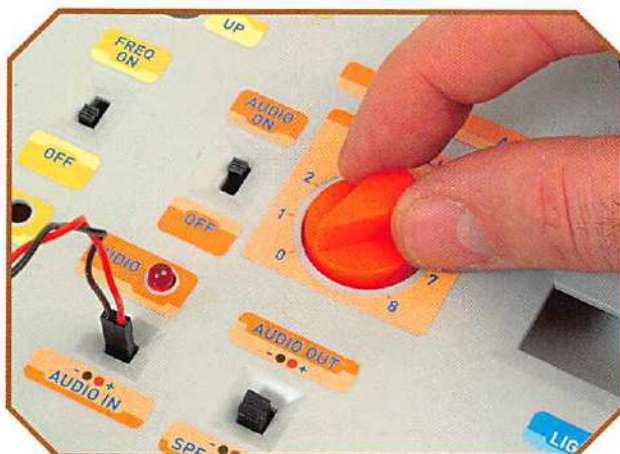


Collegamenti all'ingresso dell'amplificatore.

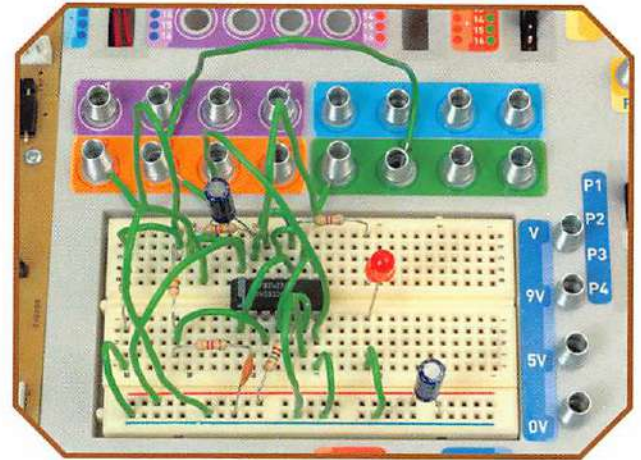
attiva. Per disattivarlo non è sufficiente collegare nuovamente l'ingresso al negativo, ma è necessario premere anche il pulsante P1, per cancellare lo stato di allarme memorizzato.

## Montaggio

Il montaggio dei componenti sulla scheda Bread Board si esegue come d'abitudine, tenendo presente l'orientamento dei circuiti integrati, quello dei condensatori elettrolitici C1 e C3 e quello del LED. Il pulsante si collega con un cavetto terminato con due connettori a due vie, da un lato lo inseriremo sul connettore siglato come P1 e dall'altro sul connettore corrispondente alle molle 13 e 14; in questo caso non è importante l'ordine della connessione, dato che il pulsante non ha polarità. Il



Comando del volume al minimo per iniziare la prova.



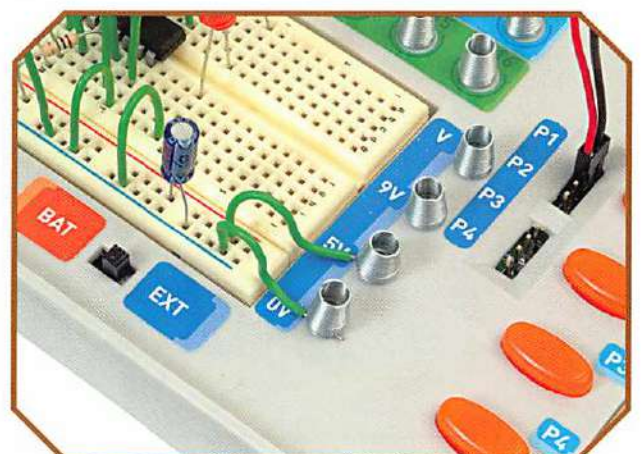
Prima di collegare l'alimentazione è necessario verificare i collegamenti.

cavetto che porta il segnale audio si collega al connettore AUDIO IN con lo stesso ordine che hanno i colori in fotografia, mentre l'altro capo del cavetto si collega ai terminali 9 e 10; il filo nero a quest'ultimo come possiamo osservare nella fotografia.

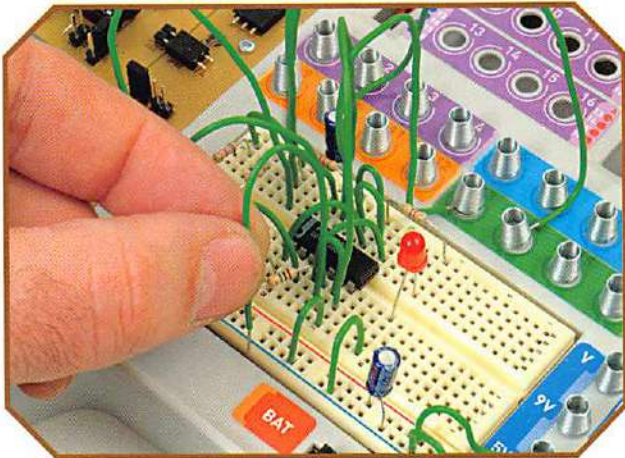
Questo circuito si alimenta a 5 V.

## Funzionamento

Dopo aver verificato tutto il montaggio, si posiziona la manopola del potenziometro del volume vicino al minimo, si collega l'alimentazione alle molle da 5 V e si preme il pulsante AUDIO ON per alimentare l'amplificatore audio. Il circuito deve rimanere in silenzio e il LED spento. Aprendo il circuito (per questo esperimento è sufficiente aprire il collega-



Collegamenti dell'alimentazione a 5 V.

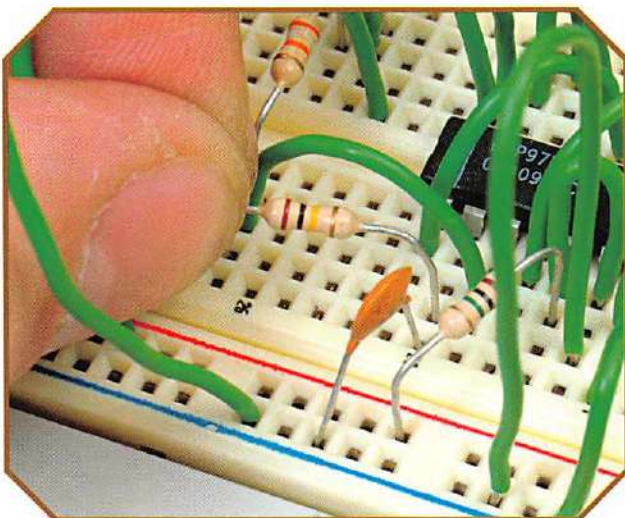


*Estrazione del collegamento per simulare l'apertura del circuito.*

mento fra i terminali 5/6 del circuito integrato e il negativo dell'alimentazione) si illumina il LED e si inizia a udire un suono sull'altoparlante il cui volume si può regolare; anche se si ristabilisce il collegamento, il circuito rimane attivo. Dopo aver collegato nuovamente l'ingresso del circuito al negativo dell'alimentazione, è necessario anche premere P1 per cancellare lo stato dell'allarme.

## Modifiche

La frequenza di oscillazione del circuito audio si può modificare cambiando i valori della resistenza R4, ma è necessario evitare di uscire dalla banda audio.



*Cambiando R4 cambia la frequenza audio.*

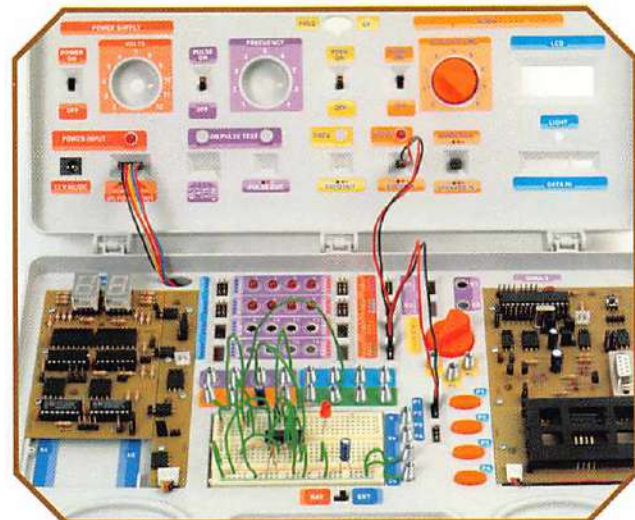


*Il pulsante P1 si utilizza come RESET.*

Il condensatore C2 e la resistenza R2 hanno il compito di applicare un reset iniziale al circuito, impedendo che si attivi collegando l'alimentazione.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
R1	Resistenza 1 M (marrone, nero, verde)
R2	Resistenza 47 K (giallo, viola, arancio)
R3,R4	Resistenza 4K7 (giallo, viola, rosso)
R5	Resistenza 39 K (arancio, bianco, arancio)
R6	Resistenza 1K8 (marrone, grigio, rosso)
C1,C2	Condensatore 10 $\mu$ F elettrolitico
C3	Condensatore 22 nF
LEDA	Diodo LED rosso 5 mm



*Esperimento completato.*



## Esercizi con la scheda Smart Card

**A**bbiamo già parlato dei differenti utilizzi che si possono fare delle schede Smart Card. Ora vedremo come lavorare con esse per caricare nuovi programmi sul microcontroller.

### Scrittura del Bootloader sul PIC

Quando si sviluppa un progetto si scrive il PIC con un programma e si monta un circuito elettrico. Se a questo circuito aggiungiamo un piccolo hardware basato su una Smart Card o su una comunicazione seriale, lasceremo aperta la possibilità di scrivere un nuovo programma sul PIC senza la necessità di estrarre quest'ultimo dalla sua sede o disporre di un hardware più completo per la scrittura. Per fare in modo che questo sia possibile, oltre alla circuiteria addizionale già citata, è necessario che sul PIC sia presente un Bootloader. Quando si avvia il PIC, questo programma cercherà di scrivere un nuovo programma di funzionamento e, nel caso non ne trovi nessuno (non ci sia la scheda Smart Card o non gliene venga fornito nessuno trami-

te la porta seriale del PC), farà partire quello che è presente nella memoria.

Per caricare il Bootloader sul PIC seguiremo la stessa procedura che abbiamo utilizzato per caricare un programma normale. Configureremo il Laboratorio per lavorare con IC-Prog, ovvero inseriremo i ponticelli su JP8 e JP9, sposteremo i ponticelli di JP1, JP2 e JP3 sulle posizioni 1 e 2 e collegheremo il cavo di trasferimento tra il PC e il Laboratorio.

Facciamo partire IC-Prog, configuriamo il dispositivo come PIC16F870 e apriamo il file "Bootloader.hex" presente sul primo CD allegato all'opera nella cartella Bootloader all'interno di Esercizi e Applicazioni.

Questo file si carica su IC-Prog con l'oscillatore, la protezione del codice e i bit della parola di configurazione già definiti. Questo perché all'interno del codice del programma sono stati definiti tali parametri. Potete vedere come fare questa definizione leggendo le informazioni allegate nella stessa cartella Bootloader.

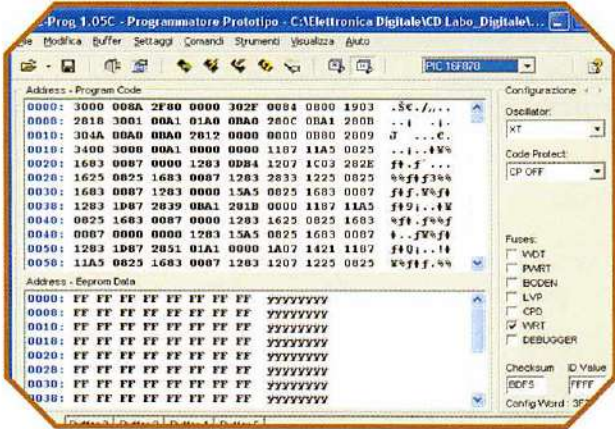
Di seguito cancelleremo il contenuto del PIC, verificheremo e scriveremo il programma mediante l'opzione Programma Tutto.

### Scrittura di un primo programma sulla scheda Smart Card

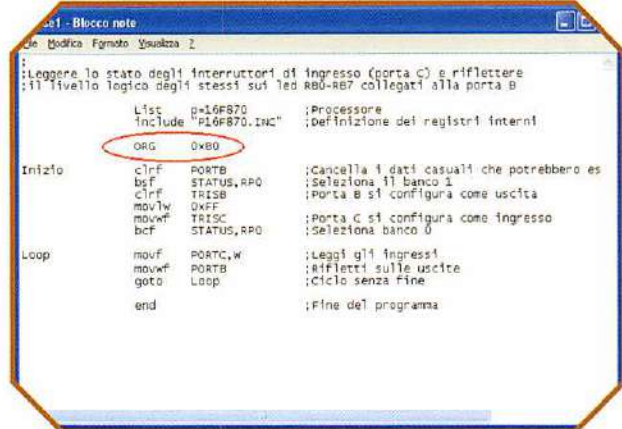
Dopo aver eseguito con successo la procedura precedente, dobbiamo caricare un pro-



Configurazione del Laboratorio per lavorare con IC-Prog.



Carichiamo Bootloader su IC-Prog.



Organizzazione del codice nella memoria di programma che deve essere corretta per lavorare con Bootloader.

gramma di funzionamento sulla scheda Smart Card.

Editeremo, per esempio, il file "ese1.asm", con il quale abbiamo già lavorato, e verificheremo che il codice occupi gli indirizzi di memoria successivi a 0xB0. Lavorando con Bootloader dobbiamo ricordare che questi occupa una precisa parte di memoria che dobbiamo evitare di sovrascrivere con i programmi di funzionamento, per questo il loro codice deve essere organizzato in modo da partire dall'indirizzo 0xB0.

Apriamo su IC-Prog il file "ese1.hex" e configuriamo il tipo di dispositivo, selezionando la memoria 24C16.

In questo caso non è necessario configurare i parametri dell'oscillatore, della protezione

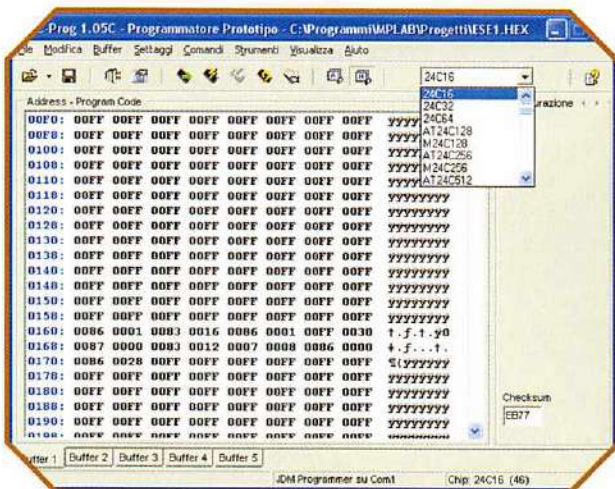
del codice e della parola di configurazione, dato che questi sono specifici dei microcontroller.

Osserviamo che caricando il file non cambia apparentemente nulla sulla videata di IC-Prog, in quanto gli indirizzi di memoria non hanno cambiato il loro valore. Ma se ci spostiamo verso altri indirizzi di memoria vedremo che il contenuto di alcuni di essi è cambiato.

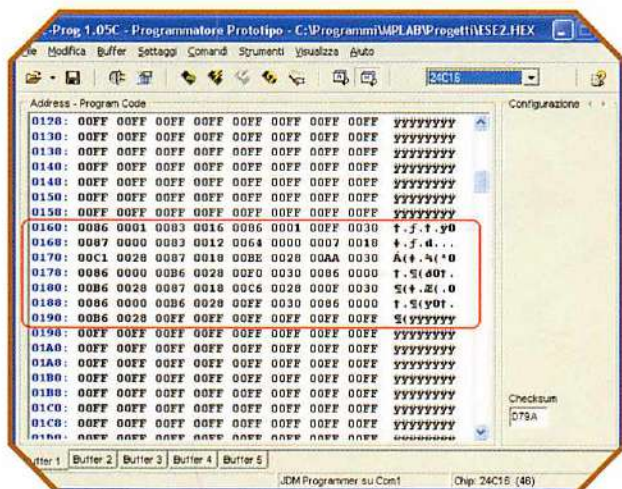
## Scrittura della scheda Smart Card

Torniamo a configurare i ponticelli del laboratorio (JP8 e JP9 estratti e JP1, JP2 e JP3 con i ponticelli su 2 e 3) per il funzionamento normale.

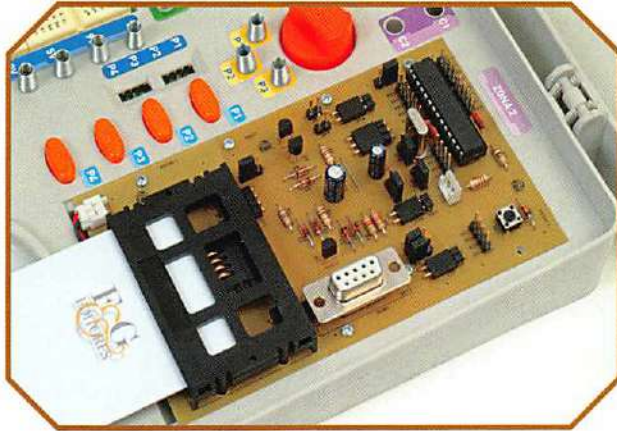
Configureremo i ponticelli JP4, JP5, JP6 e



Caricamento dell'esempio "ese1.hex".



Sulla scheda Smart Card caricheremo "ese2.hex".

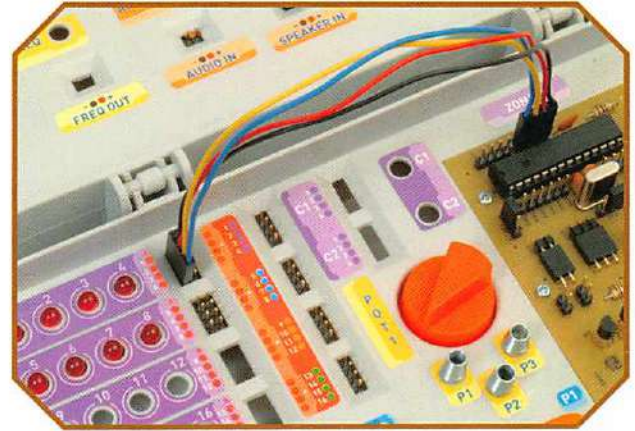


Configurazione dei ponticelli sul laboratorio per il funzionamento normale.

JP7 del Laboratorio sui terminali 1 e 2 per permettere la scrittura della scheda e inseriremo quest'ultima sullo zoccolo.

Continueremo selezionando l'opzione Programma Tutto e verificheremo che l'operazione sia stata eseguita correttamente leggendo il contenuto della memoria e confrontando quest'ultimo con il programma precedentemente inviato.

Per scrivere sulla scheda Smart Card un programma diverso si ripetono questi stessi passaggi, sarebbe utile però avere una scheda diversa per ogni programma, nel caso decidessimo di utilizzare sempre questa tecnica di programmazione, allo scopo di poter cambiare il programma del PIC in modo rapido e semplice. Dopo aver provato l'applicazione



Uniamo la porta di uscita con la matrice dei LED.

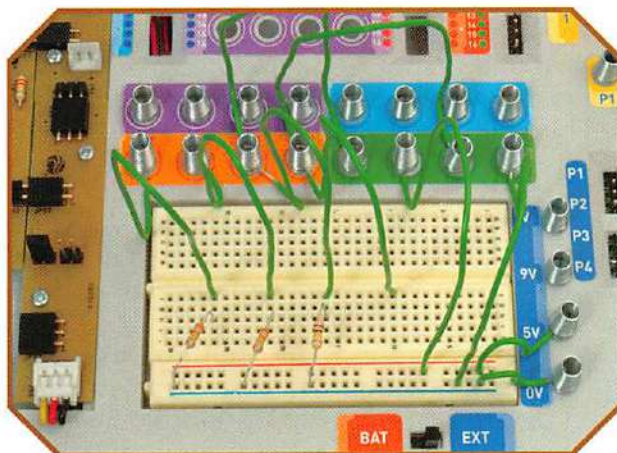
dell'esercizio ese1, caricheremo su IC-Prog l'esempio "ese2.hex".

Verificheremo, editando il codice, che anch'esso sia organizzato per funzionare con il Bootloader, ovvero che gli indirizzi che occupa nella memoria di programma inizino a partire da 0xB0, e lo scriveremo sulla Smart Card.

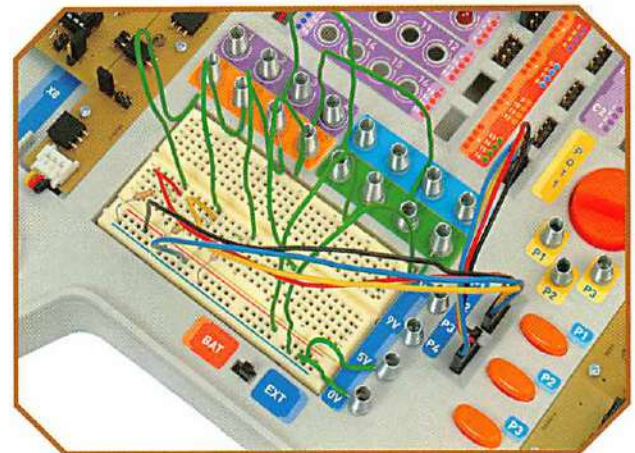
## Montaggio del circuito elettrico

Prima di eseguire il montaggio dell'applicazione dobbiamo lasciare i ponticelli dei connettori JP4, JP5, JP6 e JP7 sulle posizioni 2 e 3 e togliere la scheda Smart Card dallo zoccolo.

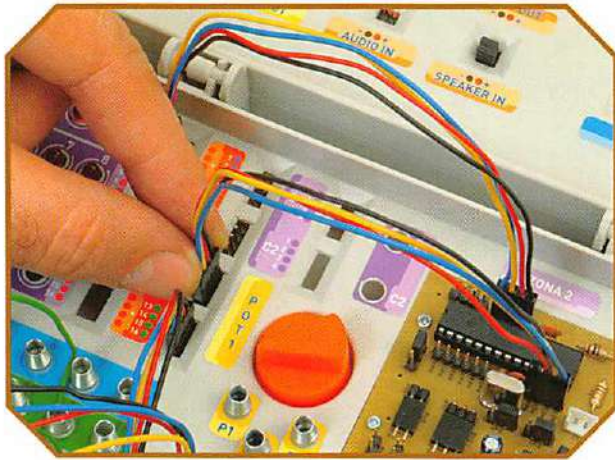
Eseguiamo il montaggio elettrico che sarà comune per entrambi i circuiti. Mediante un cavetto uniremo la porta B di uscita (RB3:RB0)



Sulla scheda Bread Board eseguiremo il montaggio degli ingressi.



Inseriamo i pulsanti tra gli ingressi e il 5 V, tranne RC3 che andrà a 0 V.

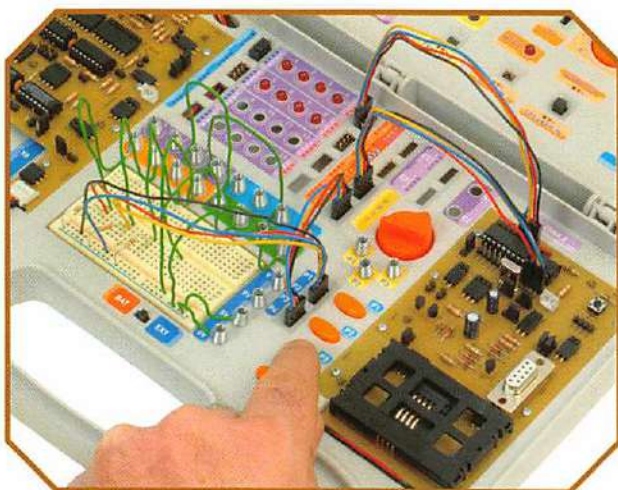


Collegiamo la porta di ingresso con le molle di collegamento.

con la matrice dei diodi LED. Ricordate che i ponticelli devono essere inseriti sui catodi dei diodi. Non utilizzeremo tutte le uscite perché sono sufficienti quattro di esse per osservare il funzionamento di questa applicazione.

Il montaggio degli ingressi si può fare utilizzando i pulsanti.

Come per le uscite, eseguiremo solamente il montaggio degli ingressi corrispondenti a RC3:RC0. Sulla scheda Bread Board inseriremo tre resistenze che da un lato verranno inserite sulle molle di collegamento arancio, e dall'altro a 0 V. Per l'ingresso RC3 eseguiremo un montaggio diverso, dato che non è necessaria la resistenza. Visto che la comunicazione con la scheda viene stabilita tramite RC3 e RC4, questi terminali non sono liberi come gli altri. Se os-



Proviamo i programmi attivando i pulsanti.

serviamo il circuito elettrico della scheda DG07 vedremo che questi terminali hanno una resistenza da 2k2 verso il positivo. Inseriremo i pulsanti tra il 5 V e il terminale di ingresso, mentre per il terminale RC3 il pulsante verrà inserito tra il terminale e 0 V.

Infine, collegheremo mediante un cavetto i terminali di ingresso RC3:RC0 al connettore corrispondente alle molle di collegamento 12:9 (colore arancio).

### Prova di funzionamento

Quando collegheremo l'alimentazione al circuito apparentemente il microcontroller non farà nulla, non si attiverà nessuna uscita anche se proveremo tutte le combinazioni di ingresso. Internamente è in esecuzione il Bootloader ma, dato che non trova nessun programma da caricare, esso si ferma.

Se, a questo punto, inseriamo la scheda nel suo zoccolo e premiamo il pulsante di reset (SW2), o togliamo e ripristiniamo l'alimentazione, si riattiverà nuovamente il Bootloader sul PIC e cercherà un programma da caricare sul microcontroller dalle sue sorgenti esterne. Dato che la scheda Smart Card è stata inserita, la leggerà e scaricherà il programma che essa contiene. Questa operazione dura circa 15 secondi.

Trascorso questo tempo il microcontroller è pronto per lavorare con il nuovo programma e indica che la comunicazione è corretta perché i terminali RC3 e RC4 sono a livello alto (in questo caso possiamo vedere che il LED corrispondente a RC3 è sempre acceso).

Possiamo togliere la Smart Card dallo zoccolo in quanto potrebbe provocare delle anomalie di funzionamento essendo collegata alle linee RC3 e RC4 del PIC.

Se abbiamo caricato ese1 potremo vedere che lo stato delle uscite corrisponde allo stato degli ingressi. Quindi se azioniamo uno dei pulsanti il diodo LED corrispondente si accenderà (con RC3 attivando il pulsante il diodo si spegnerà).

Lavorando con ese2 (dovremo ripetere gli stessi passaggi eseguiti con ese1), se attiveremo i pulsanti P1 e P2 (ingressi RC0 e RC1) vedremo che le uscite si attiveranno in funzione della tabella corrispondente a ese2 e non come avveniva con ese1.





# Controllo dei processi industriali. Il trapano: la pratica

**C**i dedicheremo ora alle fasi finali del progetto degli esercizi rivolti al controllo di processi industriali.

Realizzeremo il montaggio elettrico del progetto di automazione di una macchina per foratura, dato che per il controllo del tornio ci servirà questo stesso montaggio.

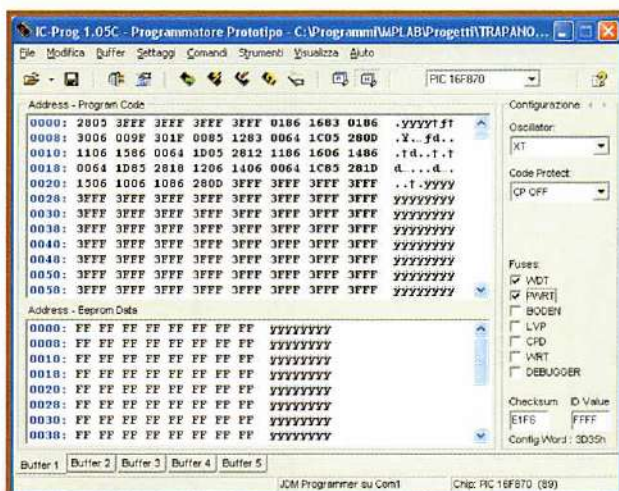
## Configurazione hardware del laboratorio per la scrittura

Anche se abbiamo visto la possibilità di caricare i programmi sul PIC mediante la scheda Smart Card utilizzando il programma Bootloader, in questo caso seguiremo i passaggi classici di scrittura del programma direttamente sul PIC.

Dobbiamo configurare il Laboratorio per la scrittura del microcontroller. Inseriremo i ponticelli sui connettori JP8 e JP9, sposteremo i jumper dei connettori JP1, JP2 e JP3 sulle posizioni 1 e 2 e collegheremo il cavo di trasferimento tra il PC e il Laboratorio.

## Scrittura con IC-Prog

È tutto pronto per stabilire la comunicazione tra il PIC e il PC, quindi possiamo far partire IC-Prog. Ogni volta che si desidera scrivere un nuovo programma sul PIC è necessario prima



Scriviamo il codice sul PIC con IC-Prog.

### PASSI PER SCRIVERE IL PIC

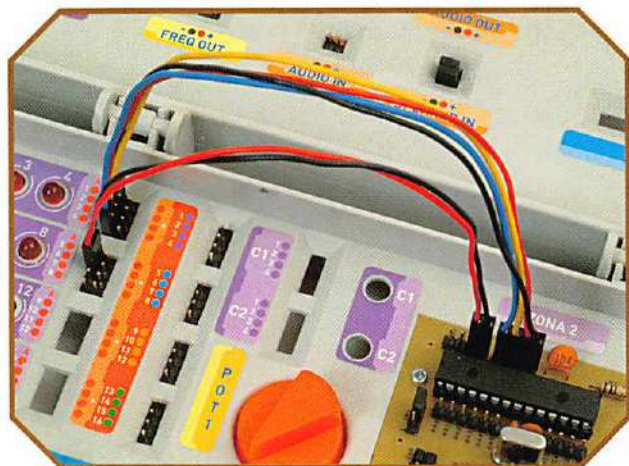
- 1 Verificare il corretto montaggio del PIC
- 2 Collegare il cavo di comunicazione Laboratorio-PC
- 3 Impostare i ponticelli su JP8 e JP9
- 4 Ponticelli su JP1, JP2 e JP3 sulle posizioni 1 e 2
- 5 IC-Prog: Fasi di lettura, cancellazione e verifica
- 6 IC-Prog: Caricamento del programma e configurazione del software
- 7 IC-Prog: Scrittura e verifica

cancellare tutto il suo contenuto. Selezionate sulla finestra a tendina della parte superiore destra il dispositivo PIC16F870, scegliete l'opzione Cancella Tutto e verificate che il PIC sia vuoto, leggendo il suo contenuto e verificando che tutti i suoi indirizzi di memoria contengano il valore 3FFF.

A questo punto possiamo caricare il codice sul PIC, quindi dobbiamo aprire il file "trapano.hex". Configureremo l'oscillatore (XT), verificheremo che la protezione del codice si trovi su CP OFF e selezioneremo i bit della parola di configurazione WDT e PWRT. Per scrivere il PIC selezioneremo l'opzione Programma Tutto



Letture del contenuto del PIC per verificare la scrittura.

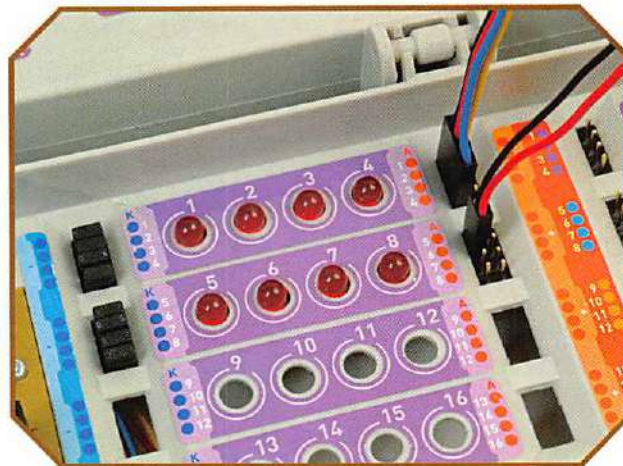


Collegamento della porta B alla matrice dei LED.

e verificheremo che venga eseguita correttamente leggendone il contenuto.

## Configurazione del laboratorio per eseguire il montaggio

Dopo aver caricato sul PIC il programma desiderato dobbiamo configurare nuovamente il laboratorio per poter montare il circuito elettrico che prova il funzionamento reale dell'applicazione. Togliamo i ponticelli dai connettori JP8 e JP9 e spostiamo i ponticelli dei connettori JP1, JP2 e JP3 sulle posizioni 2 e 3. Infine dobbiamo scollegare il cavo di trasferimento tra il PC e il laboratorio.

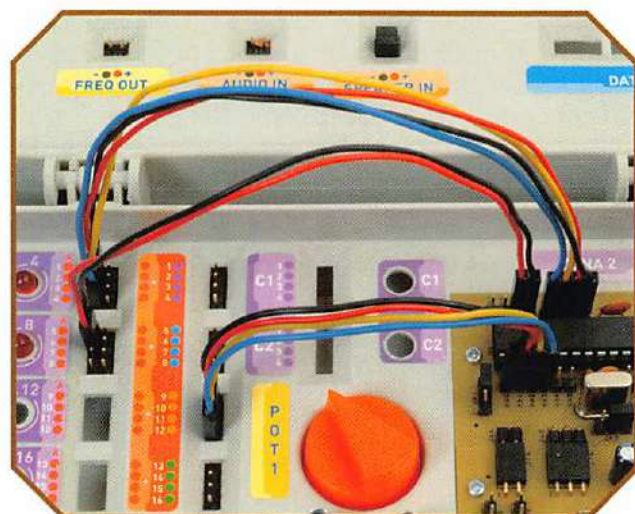


Collegamento mediante ponticelli ai catodi dei diodi.

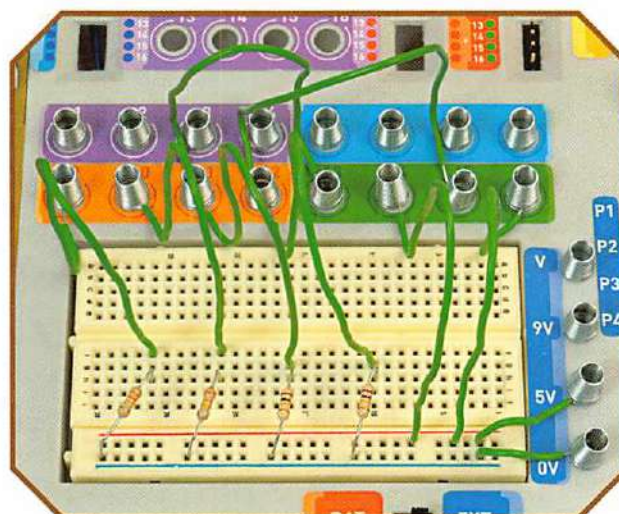
## Montaggio delle uscite

Per simulare le uscite dell'applicazione le collegheremo alla matrice dei diodi LED. In questo modo potremo verificare che le uscite rispondano alle nostre aspettative e che se al posto dei diodi collegheremo un relè o qualsiasi altro dispositivo, questo si attiverà correttamente.

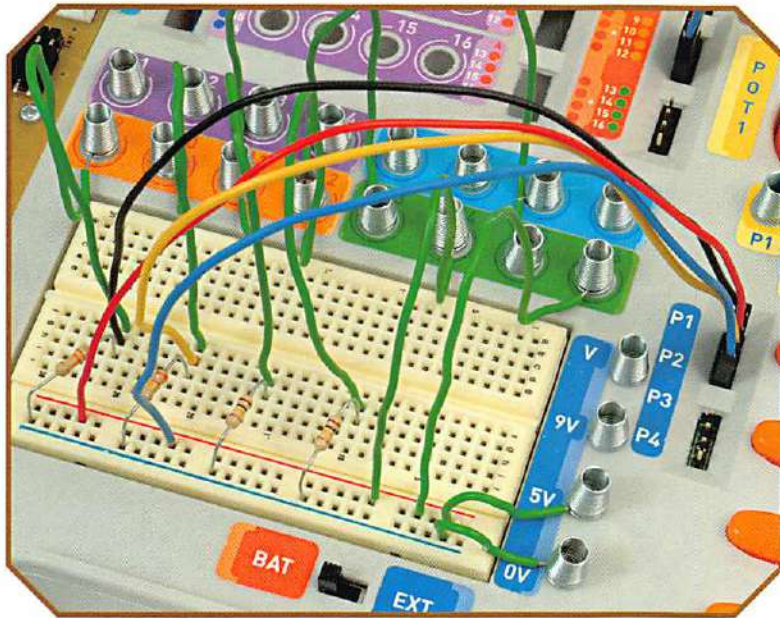
Uniamo mediante dei cavetti la porta B, che utilizzeremo come uscita, alla matrice dei diodi LED. Dato che abbiamo cinque uscite (RB4:RB0) dobbiamo utilizzare due cavetti. Ricordate sempre che quando lavoriamo con i diodi LED collegati alle uscite del PIC bisogna



Collegamento dei terminali della porta A alle molle di interconnessione.



Sulla scheda Bread Board prepariamo il montaggio degli ingressi.



Collegiamo direttamente i primi due pulsanti.

impostare i ponticelli sui catodi, come si può vedere nella figura della pagina precedente.

## Montaggio degli ingressi

Per simulare gli ingressi del sistema del laboratorio utilizzeremo i pulsanti P1:P4. Assegneremo RA0 a P1, RA1 a P2, RA2 a P3 e RA3 a P4.

Uniremo mediante un cavetto i primi quattro terminali della porta A del microcontroller con il connettore che corrisponde alle molle di interconnessione dalla 9 alla 12 (zona arancio).

Sulla scheda Bread Board inseriremo quattro resistenze i cui estremi da un lato saranno collegati al negativo, e dall'altro alle molle di interconnessione della zona arancio mediante dei fili.

A questo punto dobbiamo inserire i pulsanti, cosa che faremo in due modi differenti con due diversi tipi di cavetti. Utilizzeremo le molle di interconnessione verdi per far arrivare il positivo ai capi della resistenza che è unita ai terminali di ingresso. Queste molle ci serviranno per il montaggio

di due ingressi. Uniremo al positivo le molle 14 e 16, mentre le molle 13 e 15 le collegheremo alle molle 11 e 12 rispettivamente. Nelle figure potete vedere il montaggio che abbiamo realizzato.

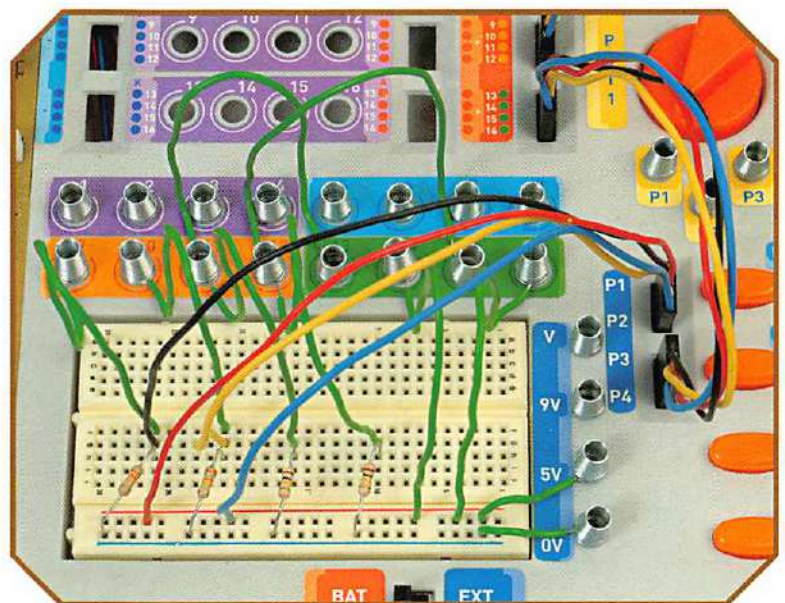
Sui due ingressi che non abbiamo ancora preparato collegheremo un filo che andrà direttamente al pulsante, quest'ultimo sarà inserito tra l'ingresso del microcontroller e il positivo (vedi figura).

Per abilitare gli altri due pulsanti dobbiamo unire mediante un cavetto il connettore di questi ultimi con il connettore che corrisponde alle molle verdi. In questo modo e con il montaggio spiegato in precedenza, abbiamo inserito il pulsante tra il positivo e il terminale d'ingresso.

## Prova di funzionamento

Con tutto il montaggio correttamente eseguito possiamo già verificare che il programma risponde adeguatamente al circuito montato.

Alimentiamo il circuito e constatiamo che inizialmente il sistema apparentemente non



Abilitiamo i due pulsanti rimanenti.

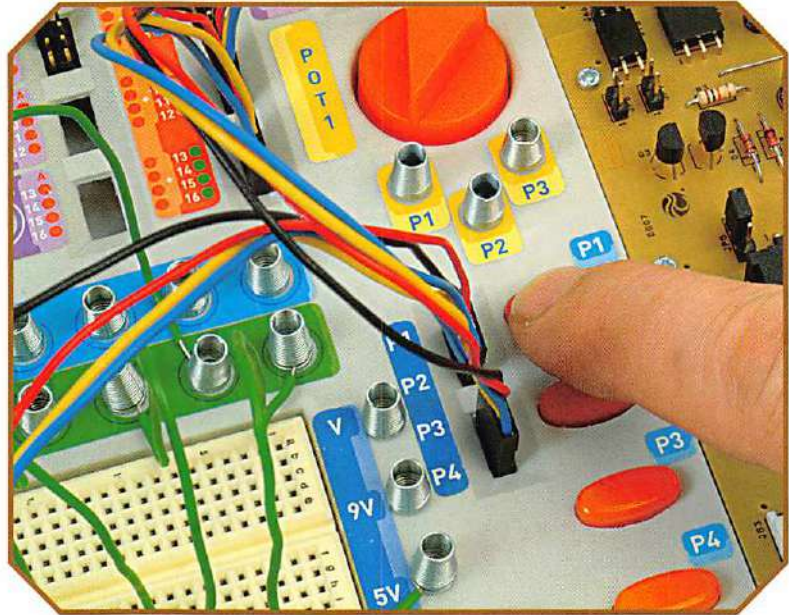


fa nulla. Premendo il primo pulsante P1 simuleremo che un nuovo pezzo stia entrando nel processo di foratura e vedremo come si attiva l'uscita corrispondente sulla matrice dei LED.

Per simulare i sensori non dobbiamo fare altro che premere i pulsanti corrispondenti. In questo modo possiamo simulare una lavorazione meccanica completa. Verificheremo che le uscite si attivino solamente in presenza delle condizioni necessarie e che il programma risponda in modo soddisfacente all'ultima fase del progetto.

### Diverse alternative da mettere in pratica

Abbiamo sviluppato il montaggio per il processo della macchina di foratura, però quando ci occuperemo del progetto del controllo di un tornio vedremo che si lavora con gli stessi ingressi e le stesse uscite, eccetto due che sul tornio non si usano. Il montaggio di questo progetto di automazione risulta quindi identico a quello già realizzato. Prova-



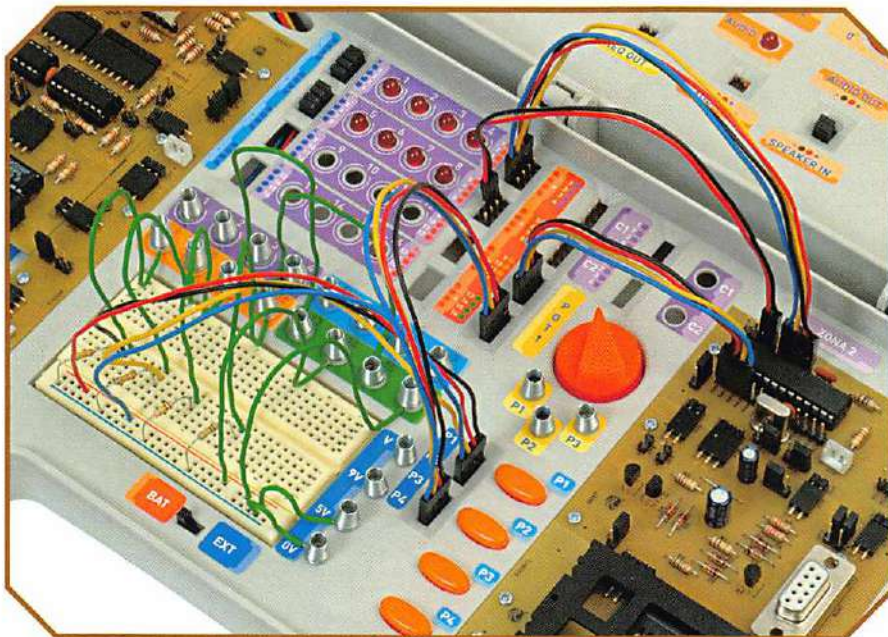
*Prova del circuito attivando i pulsanti.*

te a caricare il programma del tornio sul microcontroller e a simularne il funzionamento reale sul laboratorio.

Potete provare a modificare i programmi inserendo delle difficoltà. Immaginate, per esempio, che dopo aver attivato il pulsante che indica l'inizio del processo, questo non inizi

se non sono trascorsi 5 secondi (utilizzo di temporizzatori). Programmate un pulsante di arresto di emergenza (interrupt), o inserite qualsiasi altra modifica, dato che questo può succedere quando sviluppiamo un'applicazione per un processo industriale.

Per fare pratica potete anche caricare il programma sulla scheda Smart Card e utilizzare il Bootloader sul microcontroller per scaricare il programma. In questo modo affinerete le conoscenze e acquisirete disinvoltura e sicurezza sul lavoro con i microcontroller.



*Aspetto del laboratorio durante la simulazione.*



# Generatore di effetti sonori

**Q**uesto circuito permette di variare molti dei suoi parametri per ottenere alcuni suoni, che ci ricordano i grilli e le cicale tipici delle zone rurali.

## Il circuito

Il circuito è piuttosto originale, utilizza tre generatori di segnale e una porta OR in cui si concentrano i segnali che arrivano dai tre generatori.

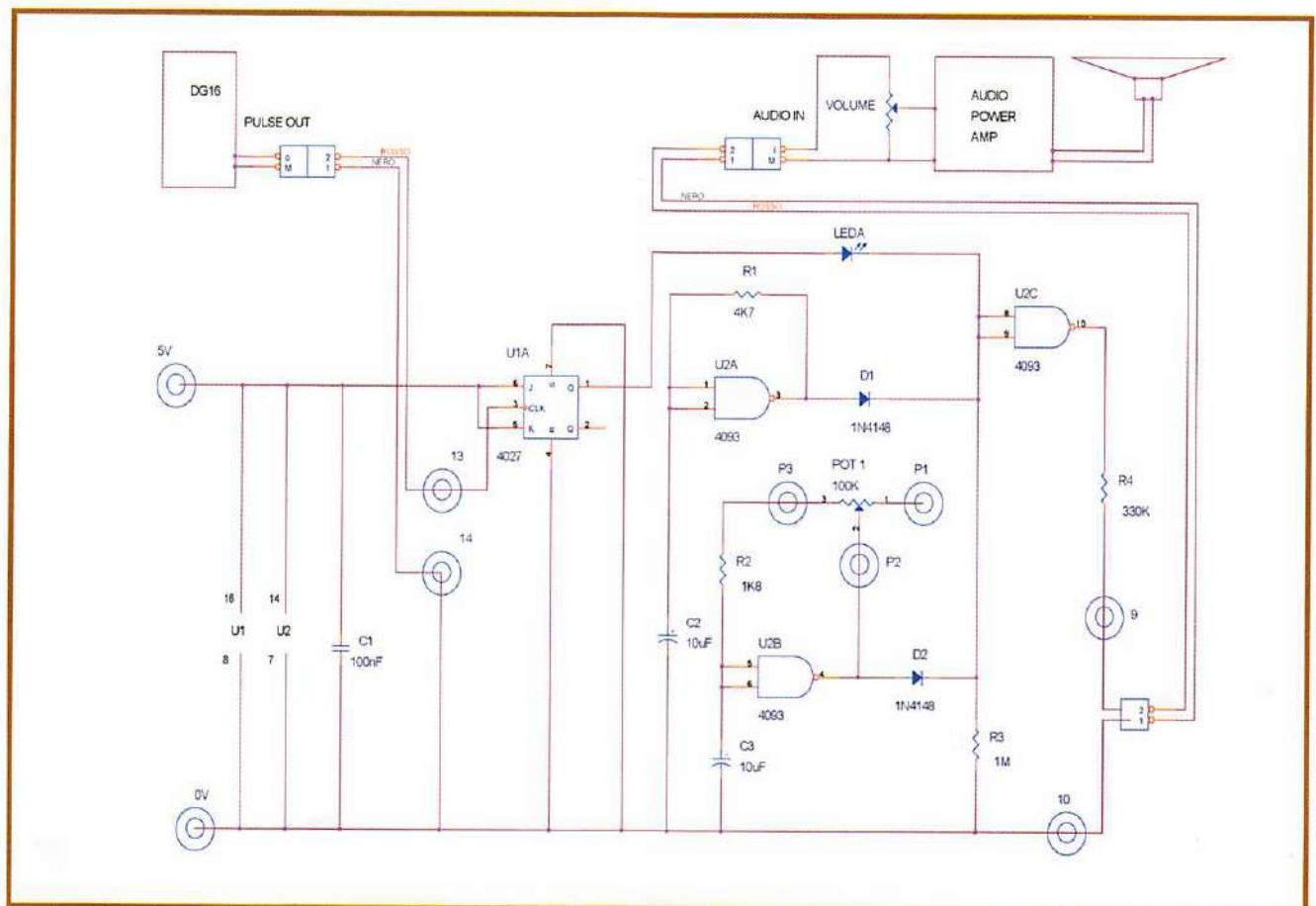
Procederemo ora a individuare la posizione di ciascun generatore.

Il più facile da identificare è quello formato dalla porta U2A, a frequenza fissa, la cui frequenza di oscillazione dipende dai valori della resistenza R1 e del condensatore C2, dopo

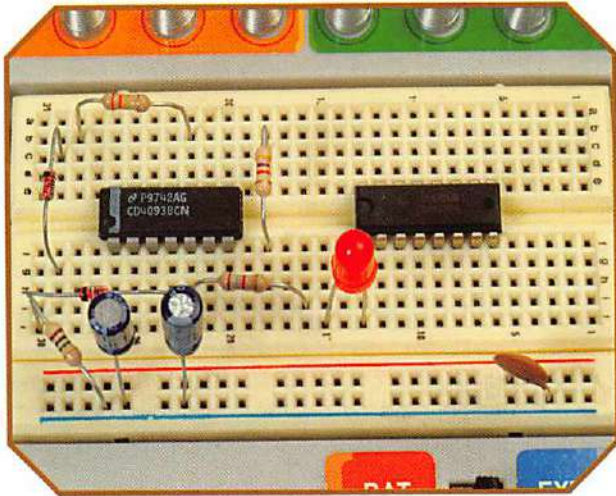
averlo montato, potrete anche provare a cambiare i valori di questi componenti.

Il secondo generatore è simile al primo, è formato attorno alla porta U2B e la sua frequenza di uscita può essere modificata ruotando la manopola di POT1.

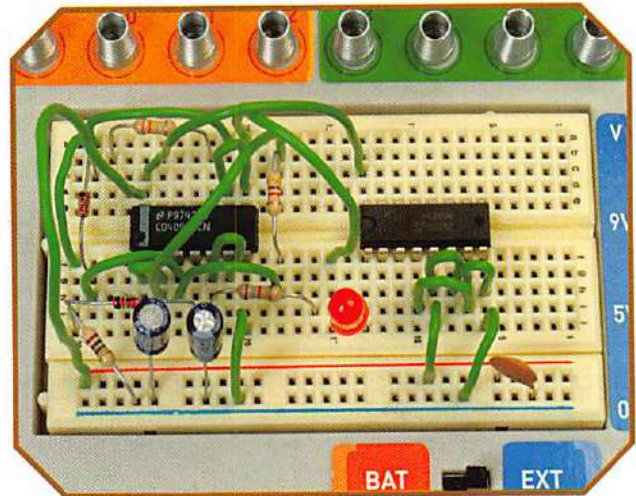
Il terzo oscillatore è formato dal generatore di impulsi e da un divisore per due, per ottenere un segnale con il 50% del ciclo di lavoro; con questo sistema è possibile ottenere una vasta gamma di frequenze ruotando la manopola FREQUENCY del generatore di impulsi.



Schema elettrico.



Componenti sulla scheda Bread Board.



Cablaggio interno della scheda.

## Porta OR

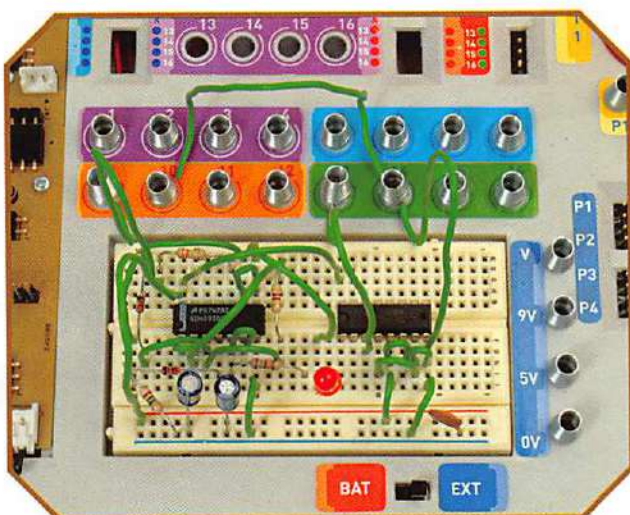
Se osserviamo lo schema vedremo che l'uscita di ogni generatore è collegata all'anodo di un diodo, mentre i catodi dei tre diodi sono uniti tra loro, in questo modo, si ottiene una porta OR. Il LED, in questo caso, si utilizza come un semplice diodo di segnale, ma dato che su di esso viene fatta circolare una corrente molto bassa, non si illumina. Il punto di unione dei tre LED si porta all'ingresso di una porta NAND che ha i suoi due ingressi uniti, quindi aggiungendo quest'ultima porta invertente alla porta OR precedente otteniamo una porta NOR.

La resistenza R4 abbassa il livello del segnale di uscita applicato all'amplificatore audio.

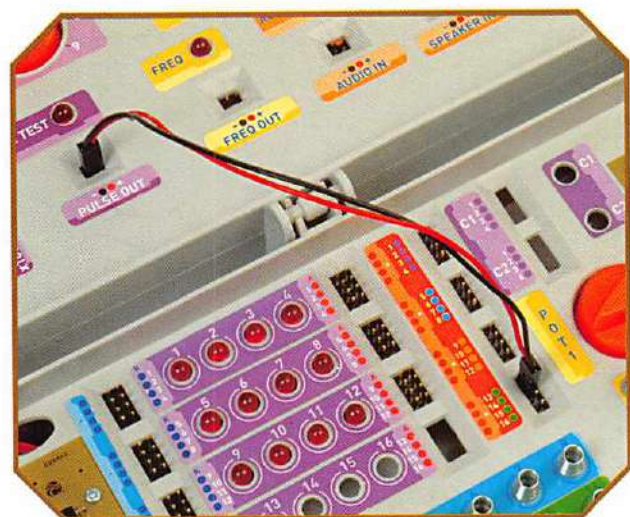
## Montaggio

Dopo aver scelto i componenti necessari si montano sulla scheda Bread Board e così si può iniziare il cablaggio degli stessi seguendo lo schema.

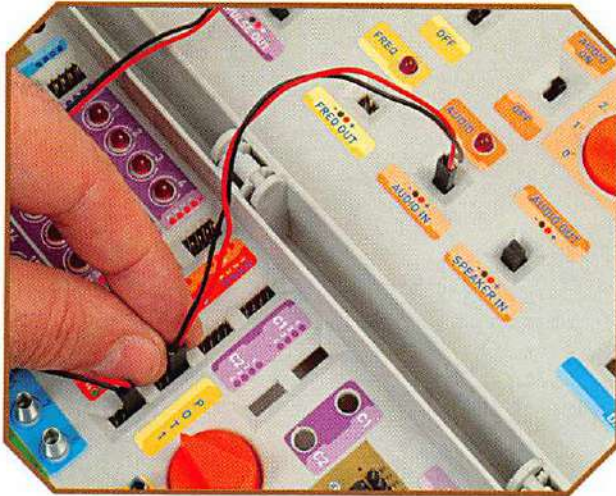
Con due cavetti a due fili si eseguono i collegamenti a PULSE OUT e a AUDIO IN rispettivamente, tenendo presente il colore di ogni conduttore, in quanto non è possibile cambiare il nero con il rosso. I collegamenti del



Collegamenti alle molle.



Collegamenti al generatore di impulsi.



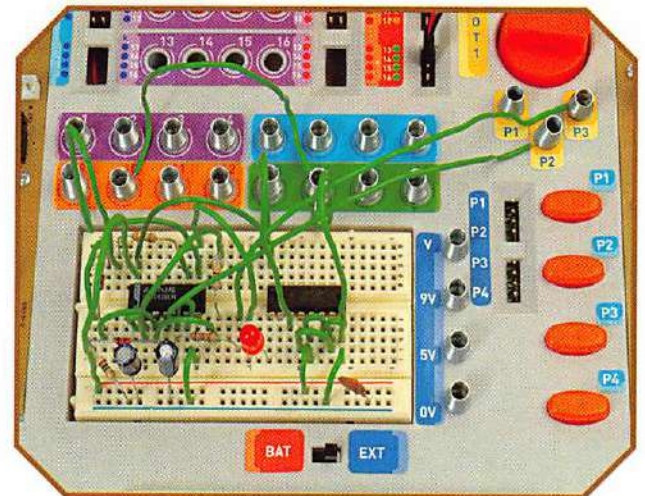
Collegamento dell'audio all'amplificatore.

potenziometro POT 1 si realizzano direttamente sulle molle. È altresì necessario collegare le alimentazioni ai due circuiti integrati e verificare che siano inseriti i ponticelli sulle uscite dell'amplificatore AUDIO OUT e sull'altoparlante.

Dopo aver verificato che ogni componente sia al suo posto e che tutti i collegamenti siano corretti si può collegare l'alimentazione.

## Funzionamento

Questo circuito può generare una grande quantità di suoni diversi. Vi consigliamo di iniziare posizionando la manopola del volume

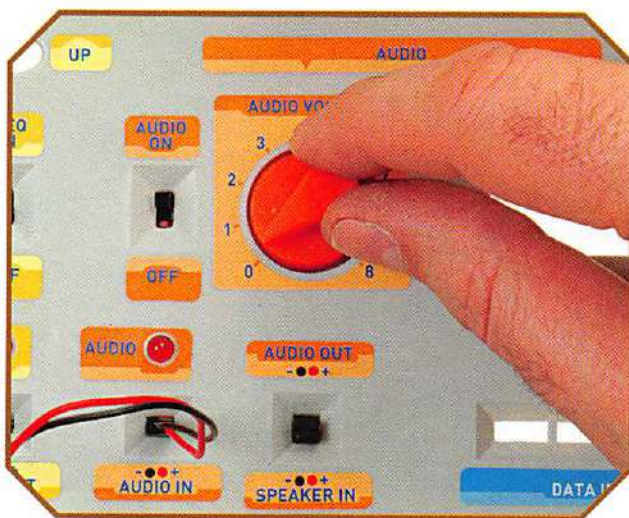


Dettaglio del collegamento alla scheda Bread Board.

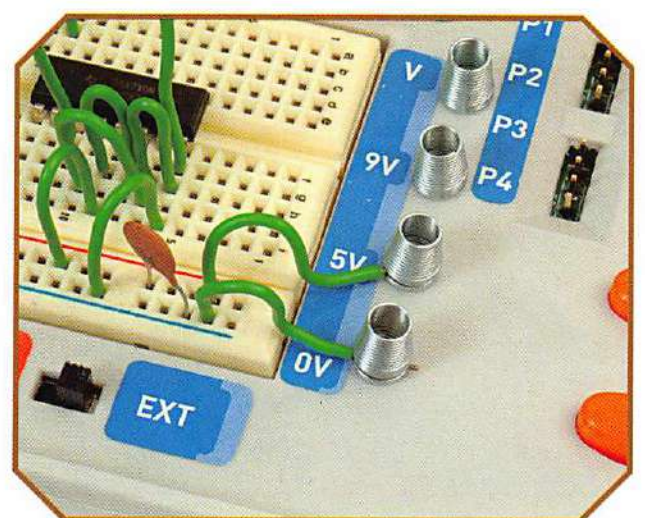
vicino al minimo, quella di FREQUENCY tra le posizioni 8 e 9, e quella di POT circa a metà della sua corsa.

Collegheremo l'alimentazione del laboratorio a un alimentatore esterno o inseriremo le batterie spostando il commutatore su BAT; sono sufficienti le pile del primo portabatterie, dato che si lavora a 5 volt (4,5 se sono batterie).

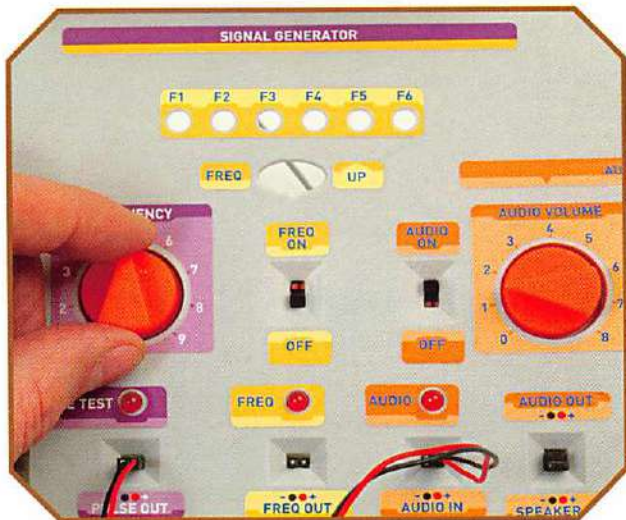
L'amplificatore si collega al commutatore AUDIO ON e il generatore di impulsi con PULSE OUT, in questo modo alimenteremo tutti i circuiti utilizzati. Ruotando la manopola di FREQUENCY e quella di POT 1 otterremo diversi effetti sonori.



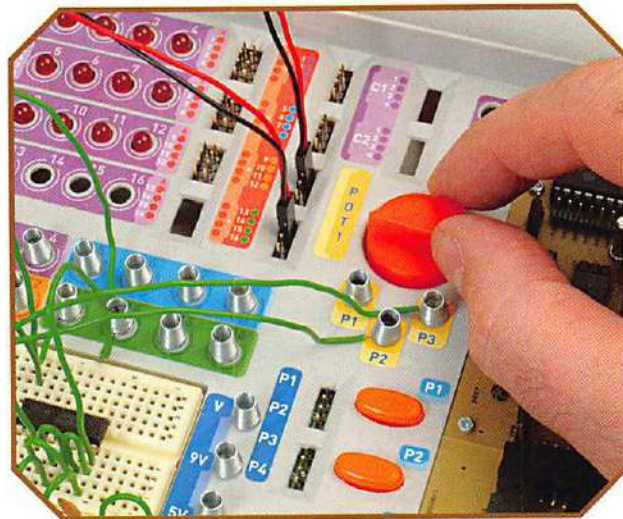
Comando del volume al minimo.



Tutto l'esperimento si alimenta a 5 V.



Tra le posizioni 5 e 9 si hanno gli effetti più evidenti.



Con POT1 si può variare anche il suono.

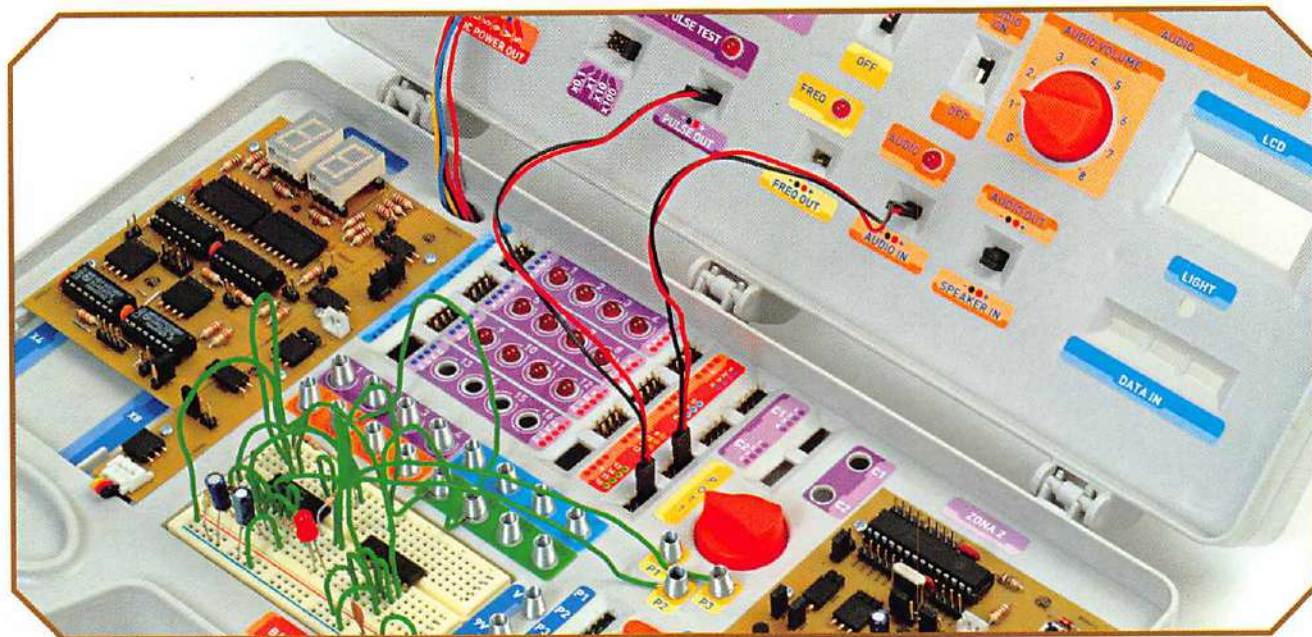
## Modifiche

Benché sia possibile ottenere una grande varietà di suoni, girando le manopole POT 1 e FREQUENCY possiamo aumentare ulteriormente le possibilità.

Se si vogliono cambiare i valori di R1 e di C2, che agiscono sul terzo oscillatore, vi consigliamo di provare per R1 con valori minimi da 1K8 fino a 10 K.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4027
U2	Circuito integrato 4093
R1	Resistenza 4K7 (giallo, viola, rosso)
R2	Resistenza 1K8 (marrone, grigio, rosso)
R3	Resistenza 1 M (marrone, nero, verde)
R4	Resistenza 330 K (arancio, arancio, giallo)
C1	Condensatore 100 nF
C2, C3	Condensatore 10 µF elettrolitico
D1, D2	Diodo 1N4148
LEDA	Diodo LED rosso 5 mm



Vista dell'esperimento completato.





# Generatore di frequenze: la pratica

**V**erifichiamo il funzionamento del nostro generatore di frequenze montando il suo circuito elettrico sul laboratorio. Eseguiremo due montaggi di uscita.

Nel primo, l'uscita sarà collegata a un diodo LED per poter vedere la risposta su quest'ultimo in base alle diverse frequenze.

Nel secondo, l'uscita sarà collegata all'amplificatore audio per ascoltare sull'altoparlante le frequenze generate.

## Scrittura del programma sul microcontroller

Per eseguire la scrittura del PIC dobbiamo prima configurare il laboratorio. Caricheremo direttamente il programma sul PIC mediante IC-Prog, quindi dovremo configurare i ponticelli JP1, JP2 e JP3 della scheda DG06, JP8 e JP9 della scheda DG07. I primi tre dovranno essere inseriti sulle posizioni 1 e 2, mentre JP8 e JP9 devono avere i ponticelli inseriti. Collegheremo il cavo di comunicazione tra il PC e il laboratorio e apriremo IC-Prog.

Su IC-Prog selezioneremo il dispositivo PIC16F870 e apriremo il file "GENERA~1.HEX". Quando si compila e si assembla il file in assembler (.asm) se ne crea uno in codice macchina (.hex) con lo stesso nome dell'originale. Se il nome è più lungo di otto caratteri utilizza la compressione del nome che abbiamo visto (~1).

Configuriamo l'oscillatore come XT, la protezione del codice su CP OFF e attiveremo i bit della parola di configurazione WDT e PWRT.

Eseguiamo la scrittura del micro e verificheremo che il processo sia stato eseguito correttamente leggendone il contenuto.

Non dimenticate che prima di scrivere il PIC è necessario cancellarne il contenuto.

## Montaggio

Ora prepareremo il laboratorio per montare il circuito elettrico che serve per provare il programma.

Scheda	Connettore	Scrittura	Funzionamento
DG06	JP1	Tra 1 y 2	Tra 2 y 3
	JP2	Tra 1 y 2	Tra 2 y 3
	JP3	Tra 1 y 2	Tra 2 y 3
DG07	JP8	Con ponticello	Senza ponticello
	JP9	Con ponticello	Senza ponticello

Configurazione dei ponticelli per i due modi di lavoro.

## Configurazione per il modo lavoro

Prima di iniziare con il montaggio dobbiamo predisporre il laboratorio in modo lavoro. L'alimentazione la riceveremo dal laboratorio stesso, quindi i ponticelli di JP1, JP2 e JP3 li potremo inserire sulle posizioni 2 e 3. Libereremo i terminali RB6 e RB7 togliendo i ponticelli da JP8 e JP9. Infine toglieremo il cavo di comunicazione.

## Montaggio degli ingressi

I terminali di ingresso con i quali selezioneremo le frequenze dell'onda di uscita sono RA2, RA1 e RA0. Sulla scheda Bread Board inseriremo tre resistenze che da un lato andranno al negativo (0 V) e dall'altro capo andranno alle molle di collegamento 9, 10 e 11. Dato che non abbiamo cavetti a sufficienza per eseguire il montaggio diretto dei pulsanti, realizzeremo un montaggio particolare per uno di essi inserendolo tra i terminali di ingresso e 5 V.



Uniremo con un filo i 5 V con la molla di collegamento 14 e con l'altro le molle 13 e 11.

Questo montaggio si può vedere nella figura. Con un cavetto a due fili uniamo il pulsante P3 con il connettore maschio corrispondente alle molle 13 e 14. Uniremo ora con un cavetto a quattro fili i terminali di ingresso RA2:RA0 con il connettore maschio corrispondente alle molle 9, 10 e 11.

Ci rimangono solamente da inserire gli altri due pulsanti tra il positivo e gli ingressi del micro. Con un cavetto eseguiamo il collegamento diretto dei pulsanti P1 e P2 alla scheda Bread Board.

Con questo montaggio abbiamo preparato tre ingressi, quindi ora passiamo a eseguire il montaggio relativo all'uscita.

## Uscita su un diodo LED

Esistono diverse possibilità all'interno del laboratorio per provare che il nostro generatore di onda quadra multifrequenza funzioni correttamente.

Il più semplice è collegare all'uscita del PIC un diodo LED e vedere se riusciamo a percepire le variazioni nella velocità di lampeggio quando variamo la frequenza mediante i pulsanti di ingresso.

Uniremo, mediante un cavetto a quattro fili, i bit meno significativi della porta B con la matrice dei diodi LED e controlleremo che i ponticelli sui catodi siano inseriti.

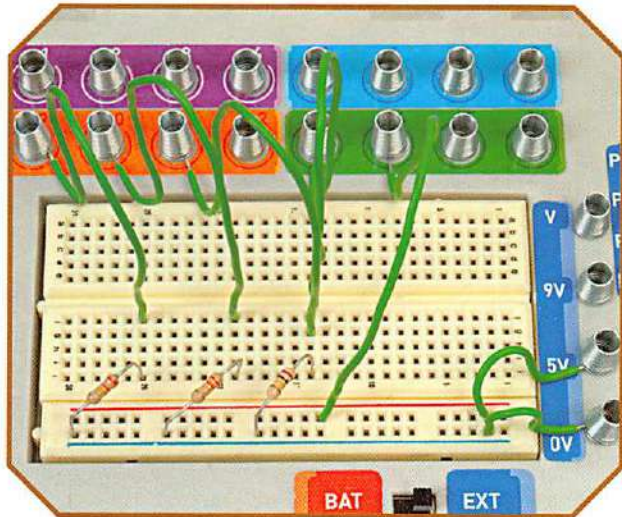
La prova di funzionamento con il LED ci permette di verificare solamente l'esistenza di tre

The screenshot shows the IC-Prog 1.05C software interface. The main window displays the program code for a PIC16F870. The code is organized into two sections: 'Address - Program Code' and 'Address - Eeprom Data'. The program code section shows a series of hex values and assembly instructions. The Eeprom Data section shows a series of FF values. On the right side, there is a 'Configurazione' (Configuration) panel with various settings. The 'Oscillator' is set to 'XT', 'Code Protect' is set to 'CP OFF', and 'Fuses' are set to 'WDT', 'PWRRT', 'BODEN', 'LVP', 'CPD', 'WRT', and 'DEBUGGER'. The 'Checksum' is '0AFC' and the 'ID Value' is 'FFFF'. The 'Config Word' is '3D35h'. At the bottom, there are buttons for 'Buffer 1' through 'Buffer 5', and a status bar showing 'JDM Programmer su Com1' and 'Chip: PIC 16F870 (89)'.

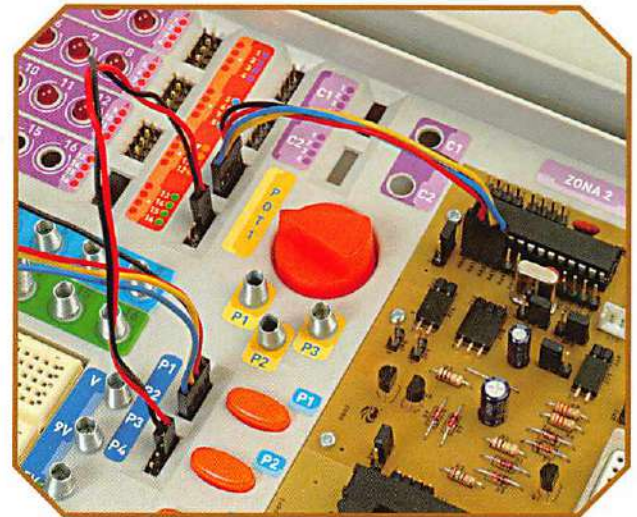
Address	Program Code
0000:	281B 3FFF 3FFF 3FFF 280E 0782 3400 3486 .yyy.,.t
0008:	34C5 34DA 34E4 34EA 34EE 34F1 00A1 0E03 AÙæîñj.
0010:	00A2 0820 0081 110B 3001 0686 0E22 0083 ç.□.t" f
0018:	0EA1 0E21 0009 0186 1683 0186 3006 009F i!.tft.ÿ
0020:	301F 0085 3001 0081 1283 3020 008B 0064 .....□f.<d
0028:	0805 3907 1D03 282E 138B 2827 2005 00A0 ....<'.
0030:	178B 2827 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF <' yyyyyyy
0038:	3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF yyyyyyyy
0040:	3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF yyyyyyyy
0048:	3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF yyyyyyyy
0050:	3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF yyyyyyyy
0058:	3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF 3FFF yyyyyyyy

Address	Eeprom Data
0000:	FF FF FF FF FF FF FF yyyyyyyy
0008:	FF FF FF FF FF FF FF yyyyyyyy
0010:	FF FF FF FF FF FF FF yyyyyyyy
0018:	FF FF FF FF FF FF FF yyyyyyyy
0020:	FF FF FF FF FF FF FF yyyyyyyy
0028:	FF FF FF FF FF FF FF yyyyyyyy
0030:	FF FF FF FF FF FF FF yyyyyyyy
0038:	FF FF FF FF FF FF FF yyyyyyyy

Carichiamo su IC-Prog il programma del generatore.



Montaggio degli ingressi sulla scheda Bread Board.



Collegamenti dei cavetti di ingresso.

stati diversi: il LED rimarrà spento quando la frequenza è 0 e la porta si trova ferma a livello "basso", il LED si accenderà completamente quando la frequenza è 0 ma la porta si trova ferma a livello "alto" e il LED sarà in uno stato di "media accensione" quando si preme un tasto qualsiasi o diversi per volta.

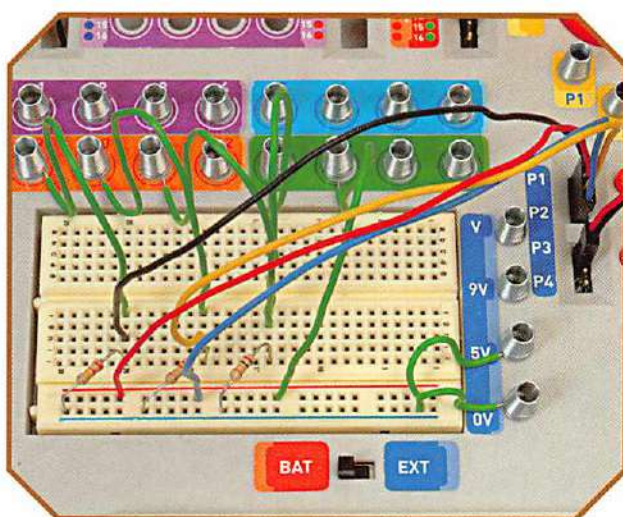
In quest'ultimo caso verranno generate delle frequenze, ma la velocità con cui cambia lo stato del LED è talmente alta che l'occhio umano non è assolutamente in grado di distinguerla. Saremo in grado di percepire unicamente maggiore o minore luminosità sul LED.

## Uscita sull'amplificatore audio

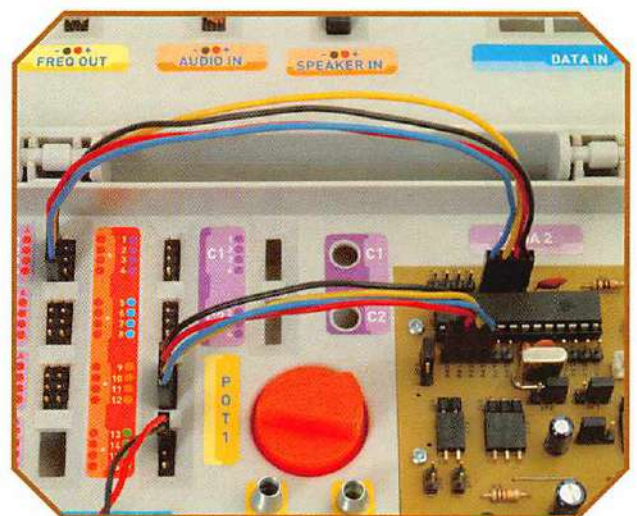
Proviamo ora un'altra delle alternative per poter osservare il funzionamento del nostro generatore.

Sfrutteremo l'amplificatore audio del laboratorio, collegheremo l'onda quadra generata all'ingresso di quest'ultimo e potremo così ascoltare come le diverse frequenze generano suoni diversi.

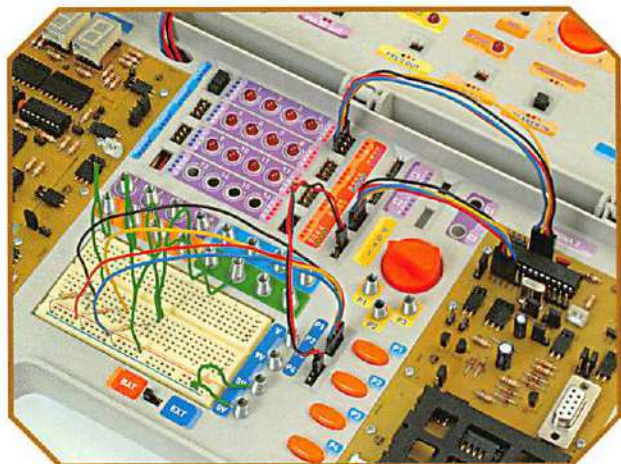
Scollegheremo il cavetto sulla matrice dei LED e lo collegheremo al connettore maschio corrispondente alle molle dalla 5 alla 8 (colo-



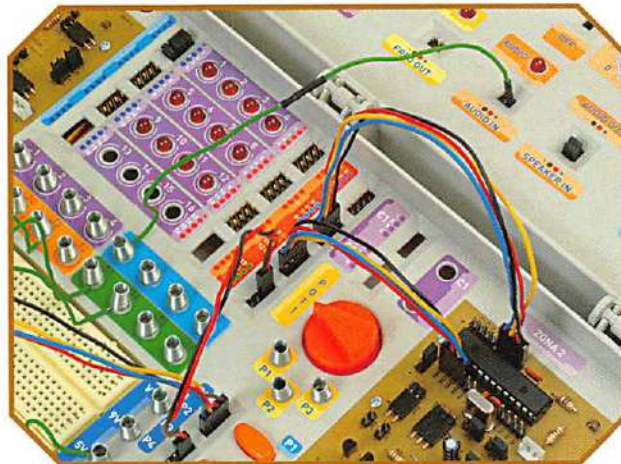
Inseriamo i due pulsanti rimanenti sul circuito.



Collegiamo l'uscita al LED mediante un cavetto.



Montaggio completo con uscita su diodo LED.



Collegiamo l'uscita del micro all'ingresso audio.

re azzurro). Alla molla di collegamento 5 arrivera, pertanto, il segnale di uscita del micro, quindi collegheremo questa molla con un filo al connettore positivo (+) dell'ingresso audio (AUDIO IN). Nell'immagine si può vedere il montaggio descritto.

Il potenziometro della parte amplificatore audio serve per controllare il volume, quindi controlleremo che non sia a 0. Inseriremo i ponticelli che uniscono i segnali di uscita audio con l'ingresso dell'altoparlante e, infine, attiveremo l'interruttore AUDIO ON.

Se proviamo le diverse combinazioni di ingresso sentiremo che i suoni generati sono diversi, dato che cambia la frequenza del-

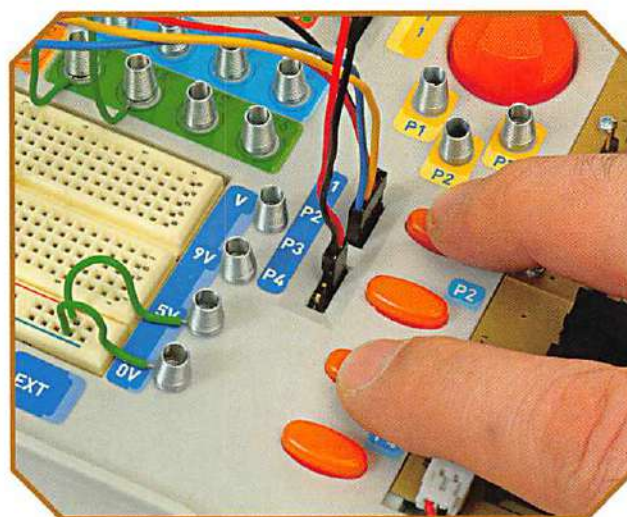
l'onda. Verificate anche cosa succede quando non ci sono pulsanti attivati.

## Conclusioni

Mediante questo programma abbiamo interagito con un nuovo dispositivo esterno, ovvero con l'amplificatore audio. Esercitatevi con il programma del generatore, ampliando il range delle frequenze (inserendo più ingressi), modificando le frequenze di uscita, ecc. Potrete verificare che i suoni ottenuti saranno diversi, che il programma è stato ben progettato e risponde in modo soddisfacente alle richieste del progetto, inoltre vi servirà per acquisire confidenza nella programmazione.



Collegamento sull'ingresso dell'amplificatore audio.



Proviamo il funzionamento attivando i pulsanti.



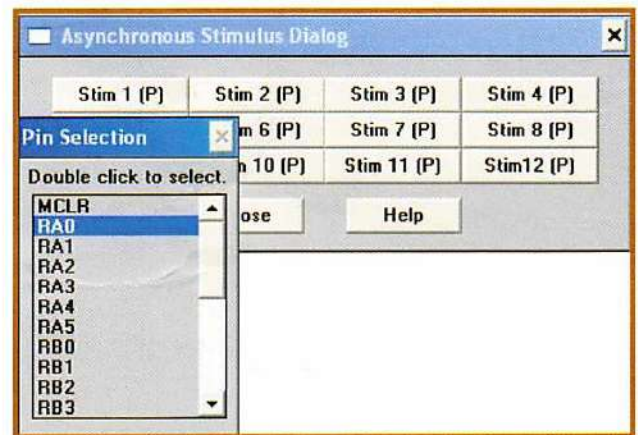
# Esercizio: "Il vostro turno", simulazione e montaggio

**C**ontinueremo il progetto della macchina de "Il Vostro Turno", simulando il programma ed eseguendo il montaggio elettrico.

Prevederemo due soluzioni di montaggio, una collegando direttamente l'uscita al display e l'altra utilizzando i driver di quest'ultimo.

## Simulazione

Facciamo partire MPLAB per verificare che il programma funzioni secondo le nostre aspettative. Dobbiamo simulare il nostro codice, e se otterremo risultati positivi, realizzeremo l'implementazione fisica dell'applicazione. Con il progetto aperto, attiveremo la finestra dei Registri delle Funzioni Speciali, una finestra con i registri PORTA, PORTB, Contatore e "w" e il Simulatore di Stimoli Asincroni, assegnando il pin RA0 al primo pulsante. Se inizieremo a simulare passo a passo premendo F7, il programma verrà eseguito correttamente. Prima di leggere l'ingresso ha letto la memoria EEPROM, e presentato sul display il digit "0", dato che sulla porta di uscita è stata portata la conversione in codice a sette segmenti di questa cifra: "00111111". Il programma si ferma nell'attesa che venga attivato il pulsante collegato a RA0. Simuleremo l'impulso assegnando al pulsante RA0 del simulatore di stimoli l'opzione "Toggle" e cliccheremo su di esso. L'ingresso si attiva e il programma salta alla routine di ritardo. Per uscire da questa routine dovremo utilizzare la finestra "Modify", scrivendo nel registro INTCON il valore "00000100", che corrisponde a "04" in codice esadecimale. In questo modo attiviamo unicamente il bit TOIF. Il simulatore uscirà dalla subroutine per tornare al programma principale e si fermerà nell'attesa che l'ingresso torni a zero per poter considerare valido l'impulso. Cliccheremo nuovamente con il mouse sul pulsante RA0 nella finestra del simulatore di stimoli "Toggle", e dato che questo



Dobbiamo lavorare con il simulatore di stimoli asincroni.

pulsante è configurato come "Toggle" (cambia di stato a ogni impulso), l'ingresso passa a livello basso "0".

Fatto questo il simulatore entra nuovamente nella routine di ritardo, dalla quale potremo uscire ripetendo i passi indicati in precedenza.

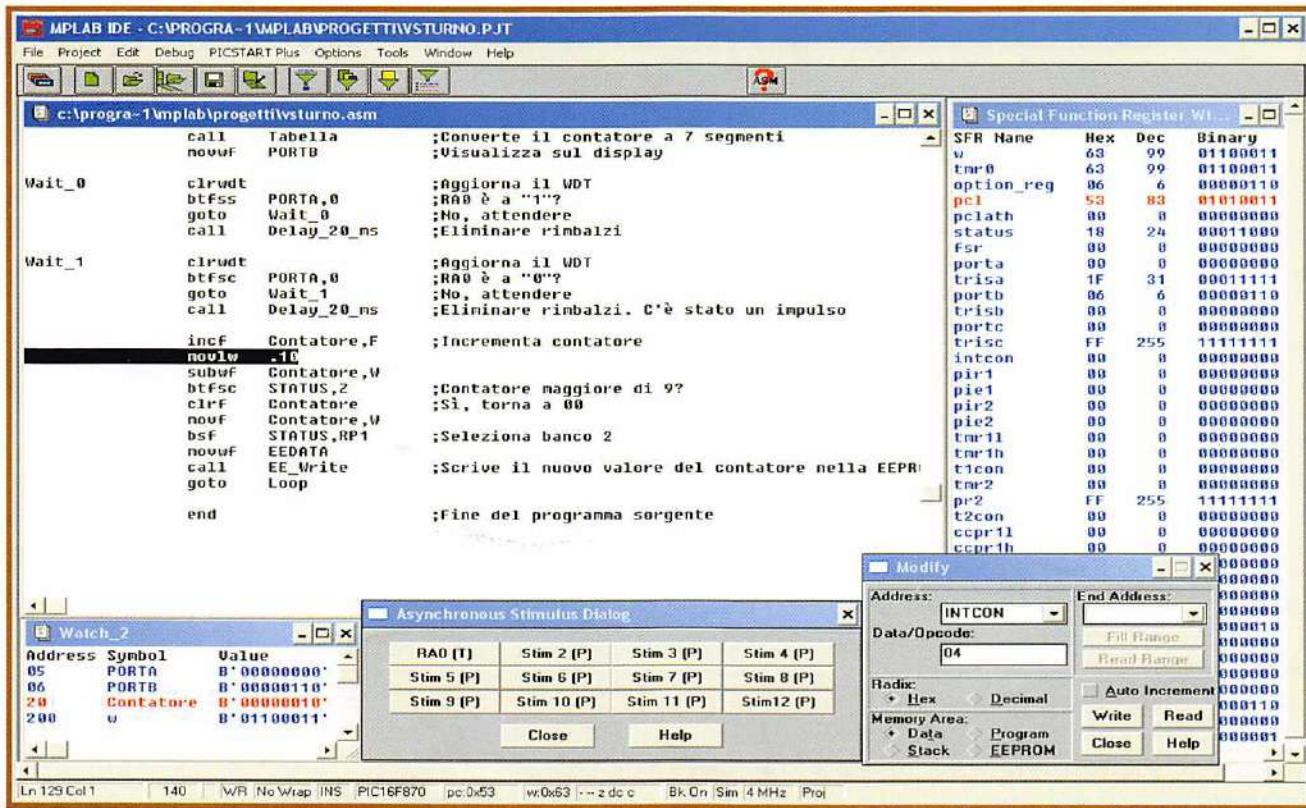
All'uscita dalla subroutine, dato che è stato fornito un impulso, dovremo vedere il contatore che si incrementa. Se continuiamo nell'esecuzione vedremo che il contatore cresce di una unità, il valore viene verificato per controllare se è maggiore di 9 e viene scritto nella memoria EEPROM facendo accesso alla subroutine di scrittura.

Il programma salterà all'inizio del ciclo, dove convertirà il valore del contatore in codice a 7 segmenti, lo scriverà sulla porta di uscita e si fermerà nell'attesa di un nuovo impulso.

Possiamo continuare la simulazione fino a

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value on: POR, BOR	Value on: all other RESET
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u

Registro INTCON che dobbiamo modificare durante la simulazione.



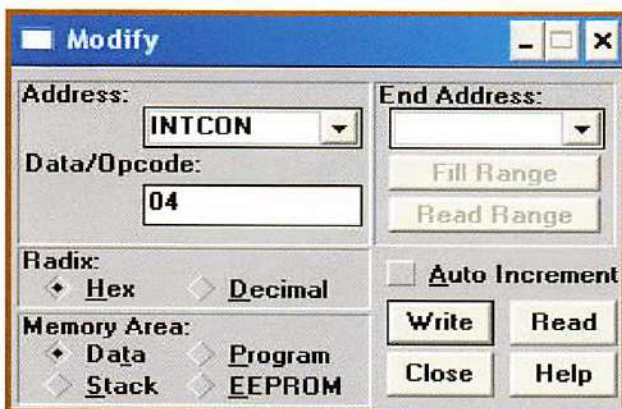
Aspetto di MPLAB durante la simulazione.

quando il contatore arriva a 9, per vedere come risponde il programma la volta successiva, ma a questo punto abbiamo già potuto verificare che il programma in simulazione funziona perfettamente.

## Preparazione del laboratorio

Bisogna scrivere il programma sul microcontroller. Configurate il laboratorio per la scrittura

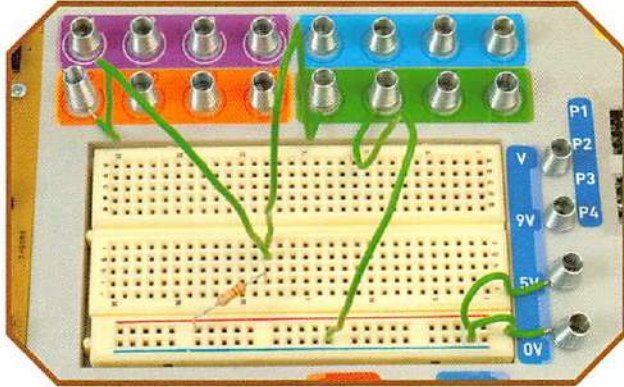
tenendo i ponticelli nelle posizioni adeguate e tramite IC-Prog cancellate il contenuto del PIC e caricate il programma, configurando l'oscillatore, la protezione del codice e i bit della parola di configurazione. Eseguite la scrittura e in seguito verificate che l'operazione sia stata realizzata correttamente. Infine, configurate nuovamente il laboratorio per lavorare in modo normale e poter montare il circuito elettrico di prova dell'applicazione.



Nella finestra Modify cambieremo il valore del bit TOIF.

## Montaggio dell'ingresso

Lavoreremo solamente con un ingresso, RA0, a cui verrà collegato un pulsante. A questo scopo, sulla scheda Bread Board eseguiremo un montaggio tipico per il lavoro con un pulsante: una resistenza che da un lato va al positivo e dall'altro alla molla di interconnessione 9 (arancio); fra questa molla e 0 V dobbiamo collegare il nostro pulsante in modo che quando lo attiveremo al pin di ingresso arrivi un livello basso di tensione (0 V). Utilizziamo le molle di collegamento 13 e 14 (verdi) per inserire il pulsante.

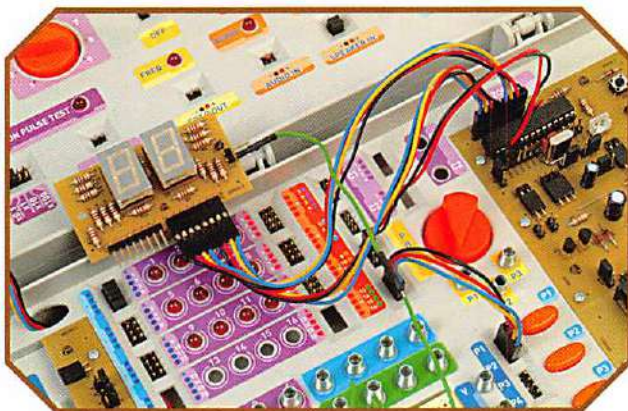


Montaggio dell'ingresso sulla scheda Bread Board.

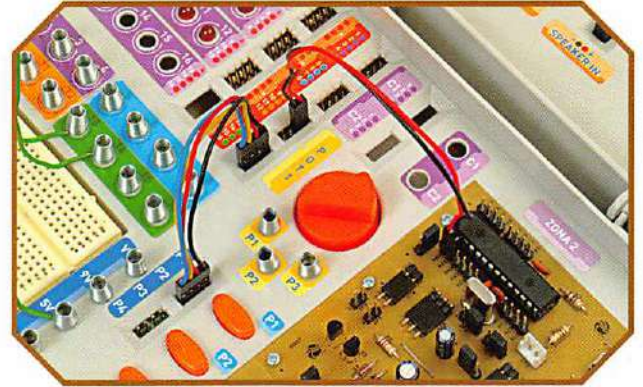
Mediante due fili collegheremo sia il pin di ingresso con la molla 9 che il pulsante con le molle 13 e 14, come possiamo vedere nella figura.

### Montaggio delle uscite direttamente al display

Dato che il nostro programma esegue una conversione fornendo sull'uscita il codice a 7 segmenti, possiamo collegare direttamente la scheda del display DG01 alla porta B di uscita. Togliamo la scheda DG01 svitandola e avvicinandola alla porta B di uscita per poterla collegare direttamente con il cavetto. Uniremo la porta di uscita a uno dei display usando due cavetti. Non dimentichiamo di unire i negativi dei due circuiti, quello della scheda display con quello del laboratorio. Utilizzeremo un filo che unisca il pin 1 del connettore J13 al segnale 0 V.



Collegamento diretto della porta di uscita al display.



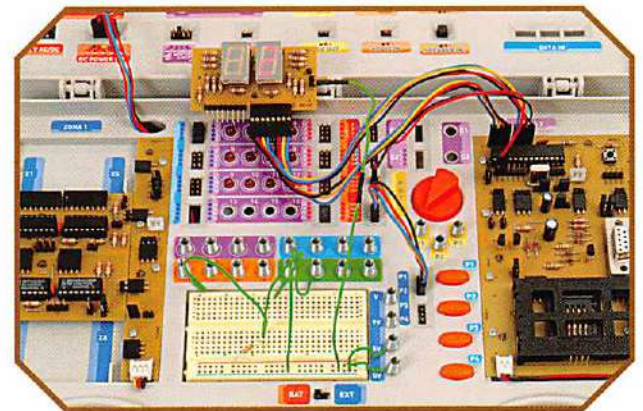
Completiamo il montaggio dell'ingresso.

Se alimentiamo il circuito inizialmente verrà visualizzato il valore "0" sul display. Premendo il pulsante P1 si incrementerà il contatore e si visualizzerà "1" sul display, e così via. Se ad un certo punto togliamo l'alimentazione, quando la ridaremo sul display apparirà la cifra che era presente prima di togliere l'alimentazione.

### Montaggio delle uscite utilizzando i driver del display

Nel laboratorio la scheda DG02 contiene i driver del display a 7 segmenti. I driver convertono un segnale binario nel codice del display. Il nostro programma fornisce sulla porta di uscita il segnale in codice a 7 segmenti, ma con una piccola modifica può fornire l'uscita in binario, così da poter utilizzare la scheda DG02.

Se sulla linea di codice alla chiamata della subroutine di conversione aggiungiamo un



Prova di funzionamento.



```

i_1      bsf    STATUS,RP1      ;Seleziona banco 2
        movf   EEDATA,W
        bcf    STATUS,RP1      ;Seleziona banco 0
        movwf  Contatore       ;Inizializza il contatore

Loop     movf   Contatore,W
        ; call  Tabella         ;Converte il contatore a 7 segmenti
        movwf  PORTB           ;Visualizza sul display

Wait_0   clrwdt                 ;Aggiorna il WDT
        btfss  PORTA,0          ;RA0 è a "1"?
        goto   Wait_0          ;No, attendere
        call   Delay_20_ms      ;Eliminare rimbalzi
  
```

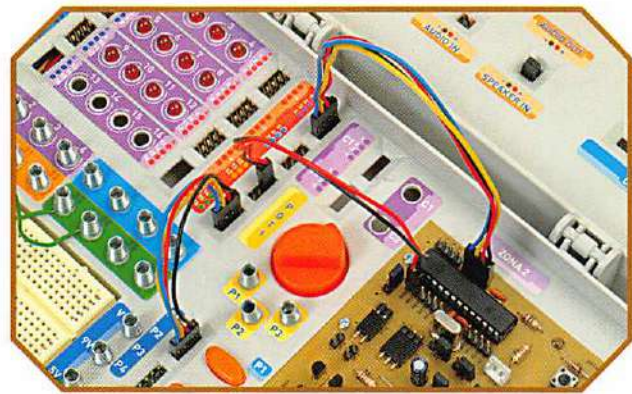
*Modifica nel programma per fornire l'uscita nel formato binario.*

“,”, come è stato spiegato quando progettavamo il codice, il segnale di uscita verrà fornito in formato binario. Dobbiamo compilare nuovamente il codice, simularlo e scriverlo sul PIC per eseguire il montaggio che vi spiegheremo di seguito.

Con la scheda DG01 nella sua posizione originale, portiamo i segnali della porta di uscita alle molle di interconnessione dalla 1 alla 4 (violetto), collegando con un cavetto la porta con il connettore corrispondente. Sono sufficienti i quattro bit meno significativi della porta di uscita, dato che in formato binario il conteggio da 0 a 9 si realizza con soli 4 bit.

Collegeremo le molle al connettore J24, che corrisponde all'ingresso del driver del display di sinistra. Utilizzeremo questo display perché il suo driver non ha gli ingressi con le resistenze, presenti invece sul driver corrispondente al display di destra.

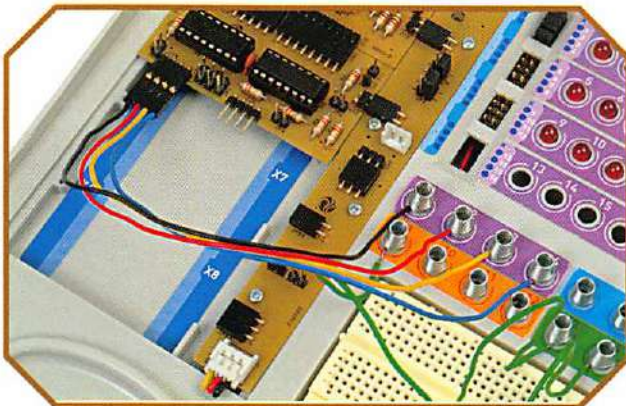
È importante non incrociare i fili e mantenere la corrispondenza fra i segnali di uscita



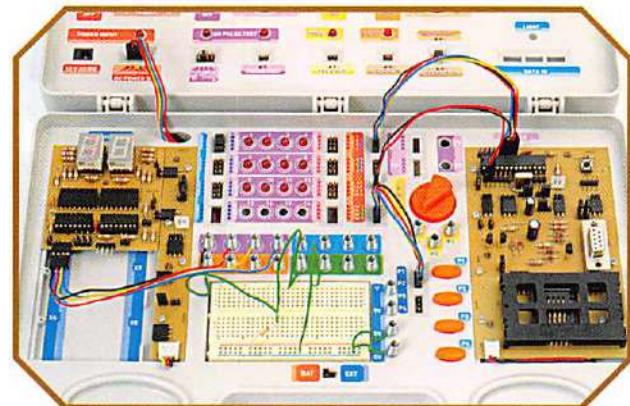
*Collegiamo le uscite alle molle di interconnessione.*

del PIC e i segnali di ingresso ai driver. Fare particolare attenzione a non incrociare i collegamenti finali e intermedi, dato che potrebbero provocare un errore di funzionamento.

Collegare l'alimentazione e controllare che il risultato fornito da questo circuito sia soddisfacente come quello del montaggio precedente.



*Uniamo le molle agli ingressi dei driver.*



*Aspetto generale del laboratorio durante la prova del circuito.*





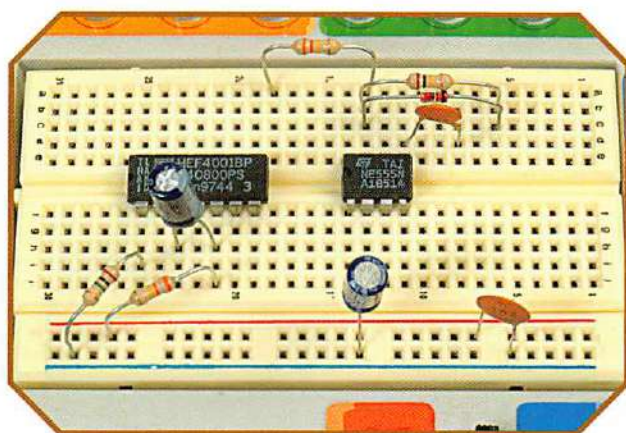
## Controllo per le barre luminose

**Q**uesto circuito si potrebbe adattare per esempio per controllare un semaforo di accesso a un garage. Si illuminano alternativamente due file di LED, mentre la terza si attiva solo per un istante dopo l'illuminazione di una delle altre.

La funzione di questo circuito è il controllo di due file di LED, che potrebbero rappresentare la luce rossa e quella verde di un semaforo, mentre l'altra fila potrebbe essere utilizzata per il giallo o per un segnale aggiuntivo, e si illumina per un determinato periodo dopo il cambio avvenuto fra le prime due.

### Il circuito

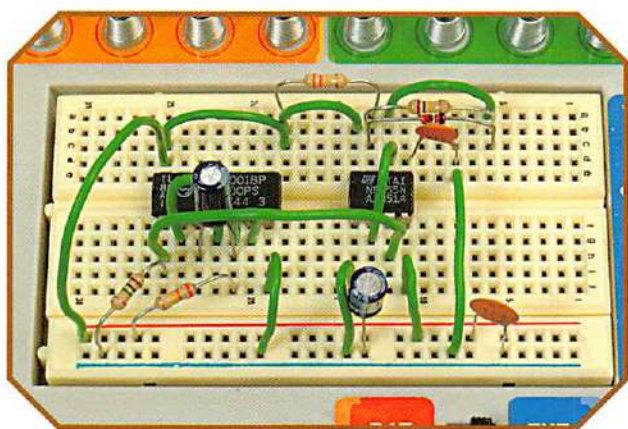
Il funzionamento del circuito è basato sul noto circuito integrato 555 configurato come astabile. L'uscita di questo circuito è a livello alto, il tempo regolato dalla resistenza R1 e dal condensatore C3, mentre il periodo per il quale l'uscita è a livello basso dipende dal condensatore C3 e dalla resistenza R2; quindi se vogliamo aumentare entrambi i tempi vi raccomandiamo di incrementare il valore di C3. Volendone cambiare solo uno e lasciare fisso l'altro, è necessario cambiare il valore della resistenza corrispondente. All'uscita di questo circuito terminale 3 di U1, sono collegati contemporaneamente otto LED, ma solamente



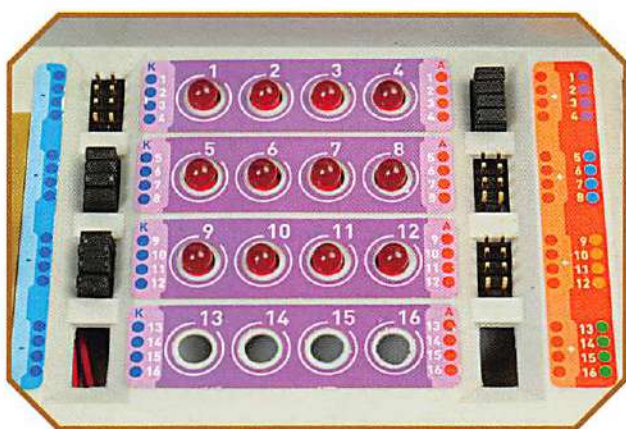
Componenti sulla scheda Bread Board.

quattro di essi si illumineranno: quando l'uscita è a livello alto si illumineranno i LED da 9 a 12, dato che i LED da 1 a 4 non sono polarizzati, cioè fra i loro terminali non c'è una caduta di tensione sufficiente per portarli in conduzione.

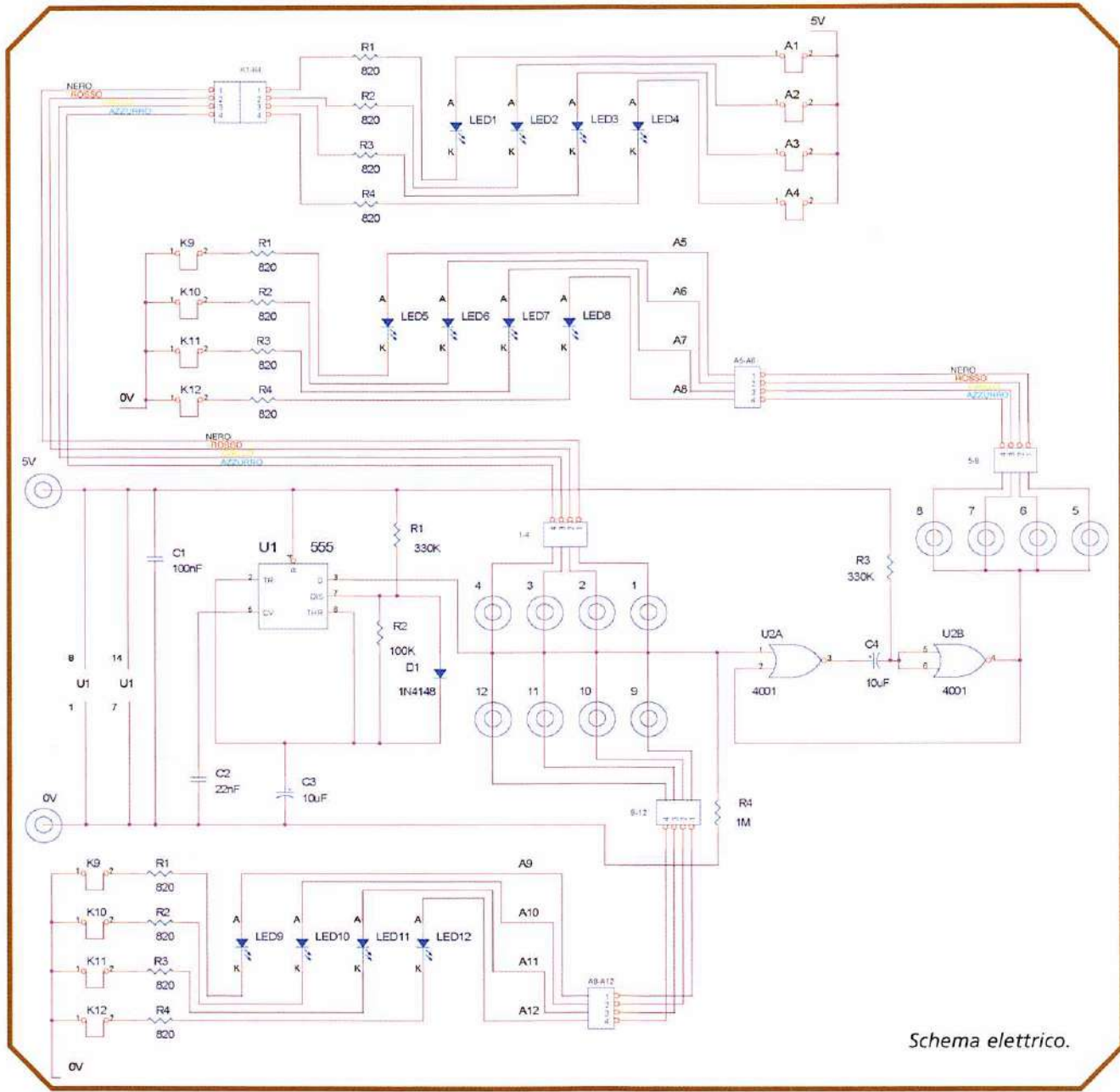
Da notare anche che questa uscita si collega all'ingresso di una porta NOR U2A. Questa porta, la U2B, il condensatore C4 e la resi-



Cablaggio interno della scheda.



Ubicazione dei ponticelli sulla matrice dei LED.



Schema elettrico.

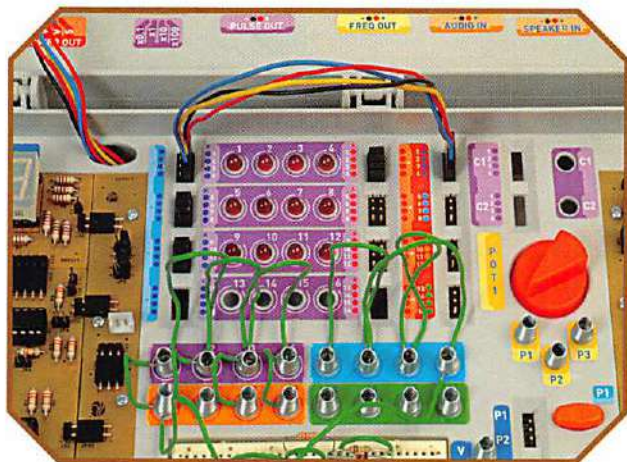
stenza R3 formano un circuito monostabile che si attiva quando si applica un livello alto all'ingresso, terminale 1, della porta U2A. Il ritardo ottenuto dipende dai valori di C4 e R3. Quando l'uscita di questo monostabile è a livello alto si illuminano i LED da 5 a 8.

### Montaggio

Il montaggio non è difficile, ma bisogna fare molta attenzione per non dimenticare nes-

sun collegamento. Inizieremo inserendo i componenti sulla scheda Bread Board, dopodiché monteremo i ponticelli sulle file dei LED, facendo molta attenzione, dato che sulla fila 1 – LED da 1 a 4 – si inseriscono sugli anodi, mentre sulle file 2 e 3 si inseriscono sui catodi.

Collegeremo ora i cavetti a quattro conduttori terminati su connettori a quattro vie. Uno dei cavetti si collega fra le connessioni dei catodi dei LED da 1 a 4, e l'altro capo ai con-

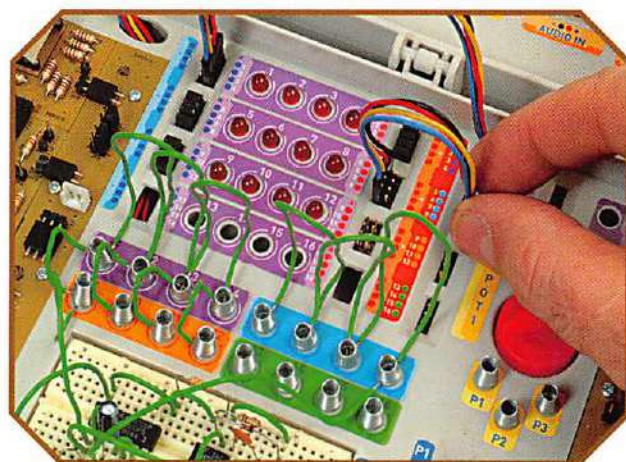


Collegamenti della prima fila di LED.

nettori delle molle dalla 1 alla 4. Il secondo cavetto si collega fra le connessioni degli anodi dei LED da 5 a 8 e le molle dalla 5 alla 8. Il terzo cavetto si collega fra le connessioni degli anodi corrispondenti ai LED da 9 a 12 e le molle dalla 9 alla 12.

### Alimentazione

Questo circuito si deve alimentare a 5 V, però prima di collegare l'alimentazione fra la scheda Bread Board e la molla che porta il 5 V è necessario rivedere con attenzione tutti i collegamenti realizzati, verificando di avere eseguito anche i collegamenti dell'alimentazione dei due circuiti integrati: per U1 il terminale 8 è il positivo e il terminale 1 il negativo, per il



Collegamenti della seconda fila di LED.

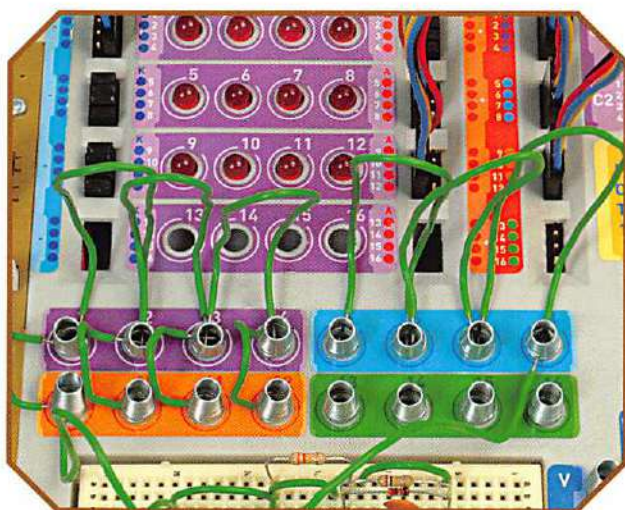
circuito integrato U2 il terminale 14 è il positivo e il 7 il negativo.

Questo circuito si può alimentare con le pile, a 5 V su posizione BAT, oppure tramite l'alimentatore esterno anche a 5 V, ma in questo caso con il commutatore dell'alimentazione sulla posizione EXT.

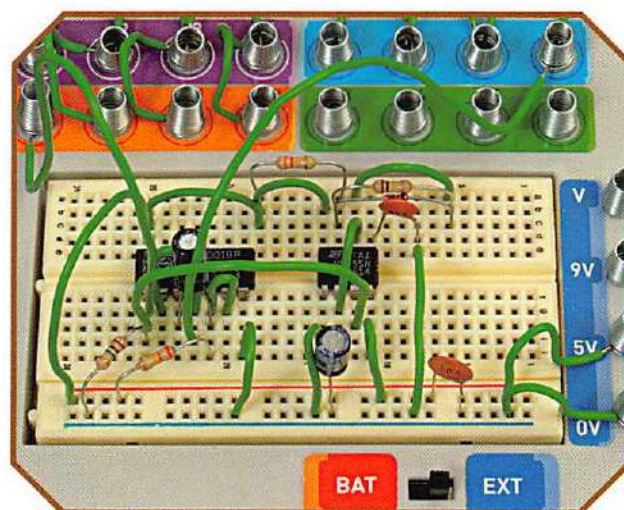
### Funzionamento

Il circuito è progettato per entrare in funzione non appena riceve la tensione di alimentazione a 5 V.

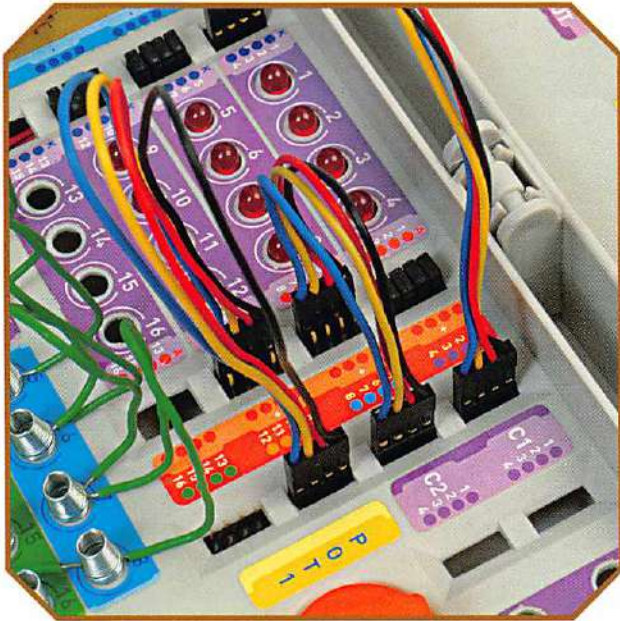
Le file 1 e 3 di LED si illuminano alternativamente, mentre la terza fila si illumina per un periodo di tempo solamente quando sull'uscita del 555 c'è un livello alto.



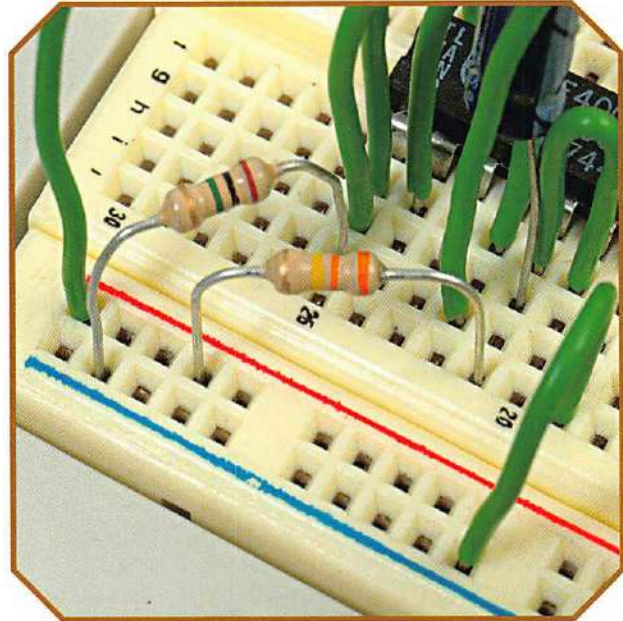
Collegamenti della terza fila di LED.



L'alimentazione deve essere a 5 V.



Dettaglio del collegamento.



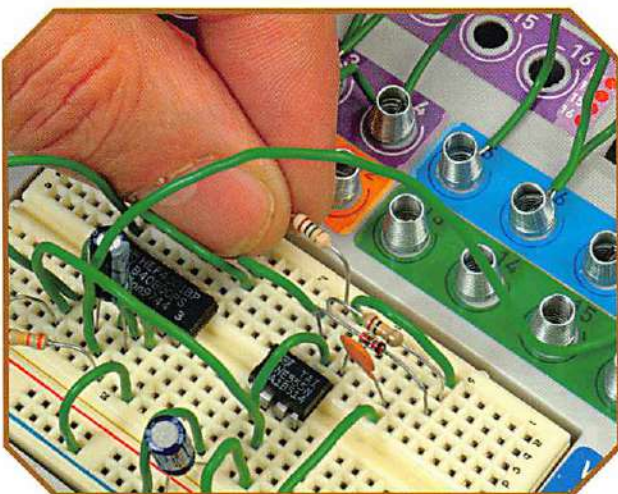
La resistenza R3 determina il tempo del monostabile.

## Modifiche

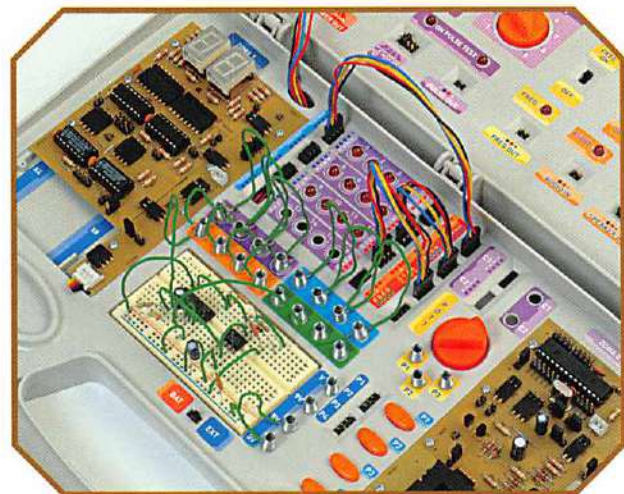
Le principali modifiche del circuito, sono quelle che hanno lo scopo di cambiare i tempi di accensione della prima fila di LED, che dipendono da R1 e da C3, il tempo di accensione della terza fila, che dipende da R2 e da C3 e il tempo del monostabile che determina il periodo per il quale rimane illuminata la terza fila, che dipende dai valori di R3 e da C4.

### LISTA DEI COMPONENTI

U1	Circuito integrato 555
U2	Circuito integrato 4001
R1, R3	Resistenza 300 K (arancio, arancio, giallo)
R2	Resistenza 100 K (marrone, nero, giallo)
R4	Resistenza 1 M (marrone, nero, verde)
C1	Condensatore 100 nF
C2	Condensatore 22 nF
C3, C4	Condensatore 10 $\mu$ F, elettrolitico
D1	Diodo 1N4148



Portando il valore di R1 a 1 M, aumenta il tempo della fila 1.



Laboratorio con l'esperimento completato.



## PicBasic Plus Lite

**L'**assembler è un linguaggio che abbiamo studiato per programmare i PIC, ma essendo un linguaggio di basso livello, utilizzarlo per risolvere applicazioni che esigono molta programmazione potrebbe risultare complicato.

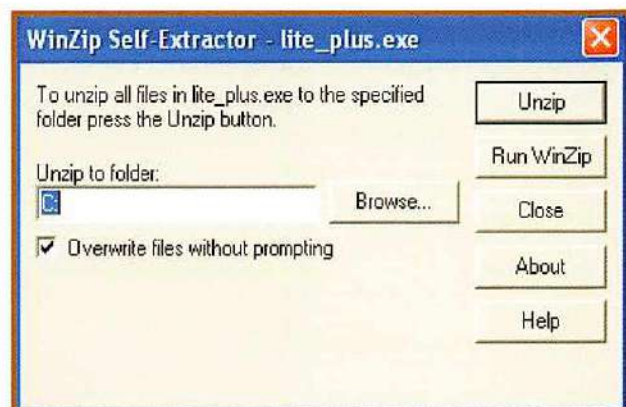
### PICBASIC PLUS LITE

Oggi è possibile trovare dei software per programmare i PIC con linguaggio di alto livello. Esistono dei compilatori e degli assembleri che convertono in linguaggi come il C o il Basic in codice macchina, per poter scrivere i programmi sul PIC.

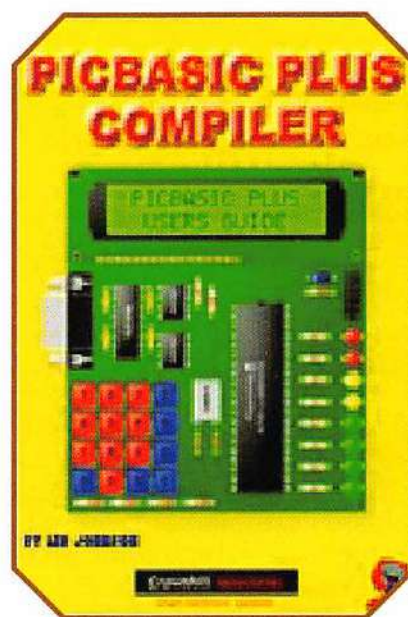
Con il PICBASIC PLUS LITE possiamo programmare in linguaggio Basic utilizzando i dispositivi del PIC16F87X, l'unico inconveniente o limitazione è che i programmi devono essere molto brevi, non devono superare le 20 linee.

### Installazione

Il PICBASIC PLUS LITE si trova sul primo CD fornito insieme all'opera all'interno della cartella di Programmi, compresso come un file ZIP di nome "lite\_plus". Per decomprimerlo cliccate due volte col tasto sinistro del mouse sul file e vi apparirà una finestra come quella della figura in basso. Dovremo inserire l'indirizzo do-



Dobbiamo decomprimere il file per poter installare il programma.



Il PicBasic Plus Lite.

ve vogliamo che il file venga decompresso e cliccare su Unzip. Quando il processo sarà terminato cliccate su Close per chiudere la finestra. All'indirizzo selezionato verrà creata una directory PBPLUS\_LITE in cui si trova il file eseguibile PBP\_Lite.

Mandando in esecuzione questo file si entra direttamente nel programma e ci verrà presentata una finestra informativa dello stesso, con la versione e le caratteristiche del compilatore. Per iniziare a utilizzare il programma clicchiamo sul pulsante OK della finestra.

### Facciamo conoscenza con il programma

Per familiarizzare col programma vedremo a che cosa servono ciascuno dei menù a cui possiamo accedere tramite il menù principale.



PICBASIC PLUS LITE

## PICBASIC PLUS LITE

**BASIC compiler for the 14-bit core devices**

Written by L. Johnson

Virtually Fully Functional Demo  
Limited to 20 lines, 2 Crystal frequencies,  
and the 16F84 - 16F877 PICmicros

✔ OK



To order the full package, contact Crownhill at  
**sales@crownhill.co.uk**

Telephone 01353 666709      Fax 01353 666710

**Crownhill Associates**  
*smart electronic solutions*

Videata iniziale del PicBasic Plus Lite.

## Il menù di File (File)

In questo menù troviamo le operazioni tipiche che si possono eseguire sui file. Possiamo creare un nuovo file (New), o aprirne uno già esistente (Open), salvare i cambiamenti eseguiti sul file (Save), salvare le variazioni come un file nuovo (Save As...) e stampare il file (Print). Per quanto riguarda le funzioni non direttamente riferite ai file possiamo configurare la stampante (Print Setup...), configurare l'editor (Editor Option) e uscire dal programma (Exit).

Nella configurazione dell'editor possiamo selezionare diverse opzioni per il lavoro con quest'ultimo (margini, numeri di linee, di pagine, ecc.) e anche scegliere il formato delle linee di codice.

## Il menù di Edizione (Edit)

Anche questo menù è tra quelli che normalmente si trovano all'interno di qualsiasi programma. In esso troviamo le funzioni di Edit come quella di ripetere gli ultimi cambiamenti (Redo) o annullare le ultime modifiche (Undo), tagliare (Cut), copiare (Copy), incollare (Paste) o selezionare tutto (Select All), trovare un testo (Find...) o sostituirlo (Replace...) e infine, una funzione per passare da

minuscolo a maiuscolo (To UpperCase) e viceversa da maiuscolo a minuscolo (To LowerCase).

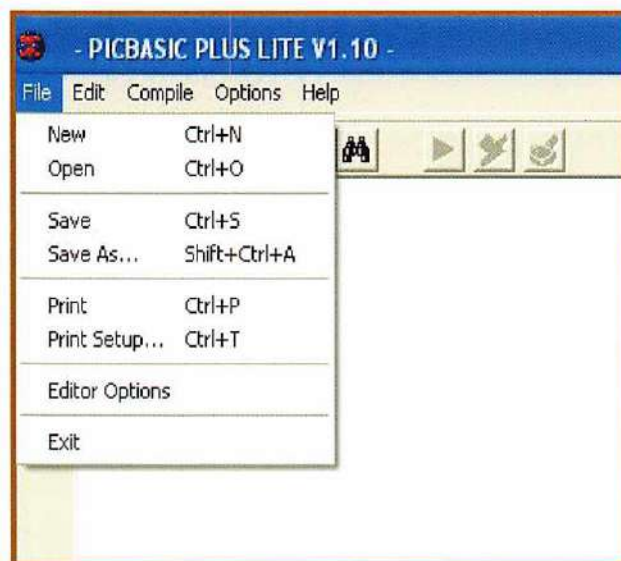
## Il menù di compilazione (Compile)

Questo menù è specifico dei programmi di programmazione. In esso si contemplanano le opzioni di compilazione del programma. Mediante Compile Basic si esegue la compilazione del codice che è stato confezionato nell'editor, ovvero si tradurrà il programma scritto in Basic in linguaggio assembler. Da questo processo, quindi, otterremo il file in assembler ".asm", il file in esadecimale da trasferire al microcontroller ".hex", il file con il listato del codice ".cod" e quello che riporta gli errori rilevati nella compilazione. Viene anche creato un file proprio del programma con estensione ".pbp". Tutti questi file vengono generati nella directory Samples all'interno della cartella del programma PBPLUS\_LITE.

Mediante l'opzione Pic Programmer potremo scrivere sul microcontroller dopo aver compilato il programma. Il programmatore lo dobbiamo scegliere nel menù che spiegheremo di seguito.

## Il menù delle Opzioni (Options)

La prima opzione di questo menù Programmer Options serve per scegliere il programma-



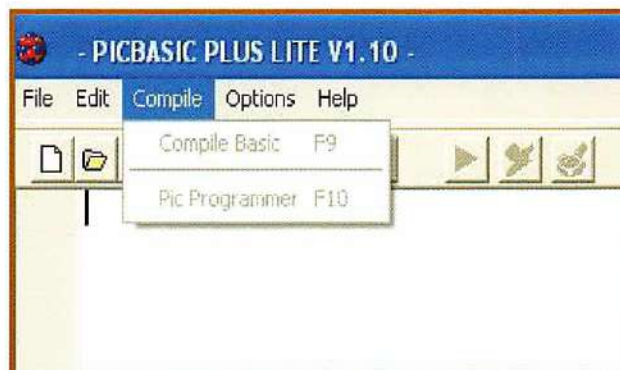
Funzioni del menù File.



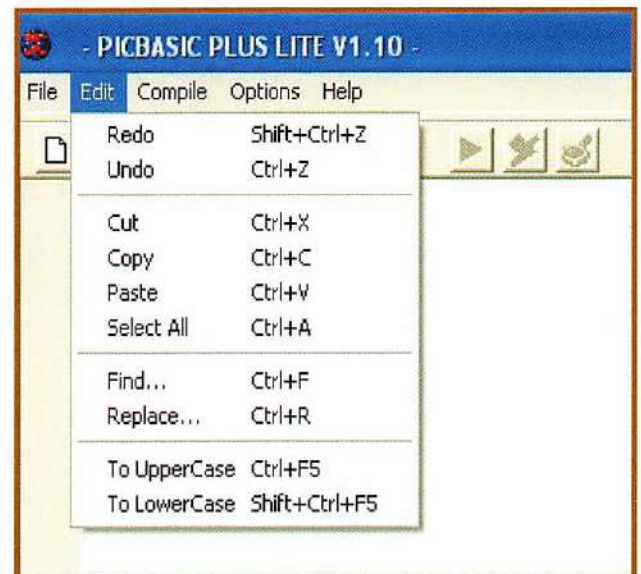
tore o l'elemento hardware che permetta di scrivere il nostro programma, una volta compilato, sul microcontroller. È possibile eseguire una scelta fra tre opzioni, anche se nel nostro caso è indifferente qualsiasi sia la scelta, dato che utilizzeremo il software IC\_Prog come software esclusivo di scrittura. Le due opzioni successive Change Epic Details e Change External Programmer Details servono per selezionare l'indirizzo dove risiedono i programmi che fanno riferimento a questi programmatori, ma neanche questi rivestono molta importanza. Le opzioni successive, invece, ci possono essere molto utili. Show ASM Window e Show HEX Window mostrano entrambe una finestra con il codice in assembler (asm) generato e con il codice in linguaggio macchina (hex). In questo modo potremo vedere l'equivalenza tra i tre linguaggi, benché per il linguaggio macchina sia piuttosto difficile capire qualcosa del programma, in quanto tutte le istruzioni sono state trasformate da mnemonico (sia esso del Basic o quello dell'assembler) in una sequenza di caratteri alfanumerici. L'ultima opzione di questo menù è Calcolatore, che è un accesso diretto alla calcolatrice interna di Windows. È utile avere una calcolatrice a portata di mano per eseguire i calcoli delle temporizzazioni, delle conversioni tra formato binario ed esadecimale, o qualsiasi altro calcolo di una certa complessità.

## Il menù di Aiuto

Questo menù presenta solamente due opzioni, Syntax e About. La prima opzione apre il manuale del programma LetPicBasicPlus su Internet Explorer. Ha un formato ".htm" ed è un



Il menù di Compilazione.

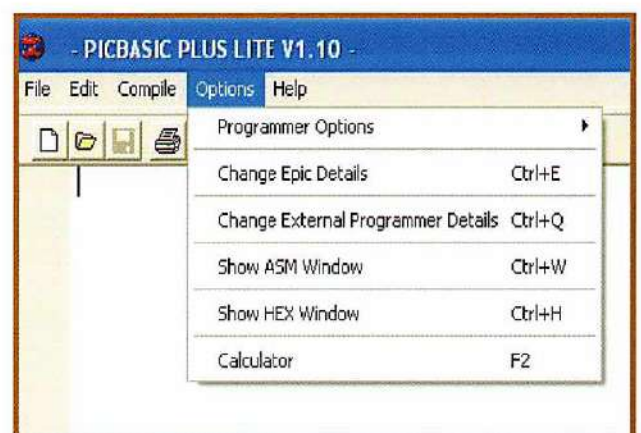


Funzioni del menù Edit.

manuale completo orientato principalmente alla programmazione in Basic interrelazionata con le funzioni del microcontroller. La seconda opzione apre la videata iniziale del programma dove appare la versione del compilatore.

## La barra degli strumenti

Le icone della barra degli strumenti riassumono le funzioni più importanti tra quelle che abbiamo appena visto. Posizionando il cursore del mouse su una di queste icone viene visualizzato un testo che ne descrive la funzionalità.



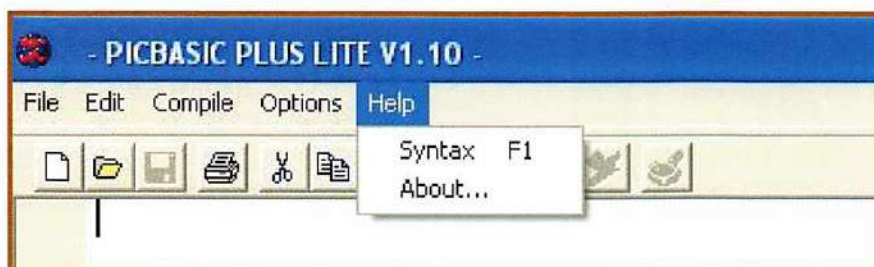
Il menù di Opzioni.



Barra degli strumenti.



Il menù di Aiuto.



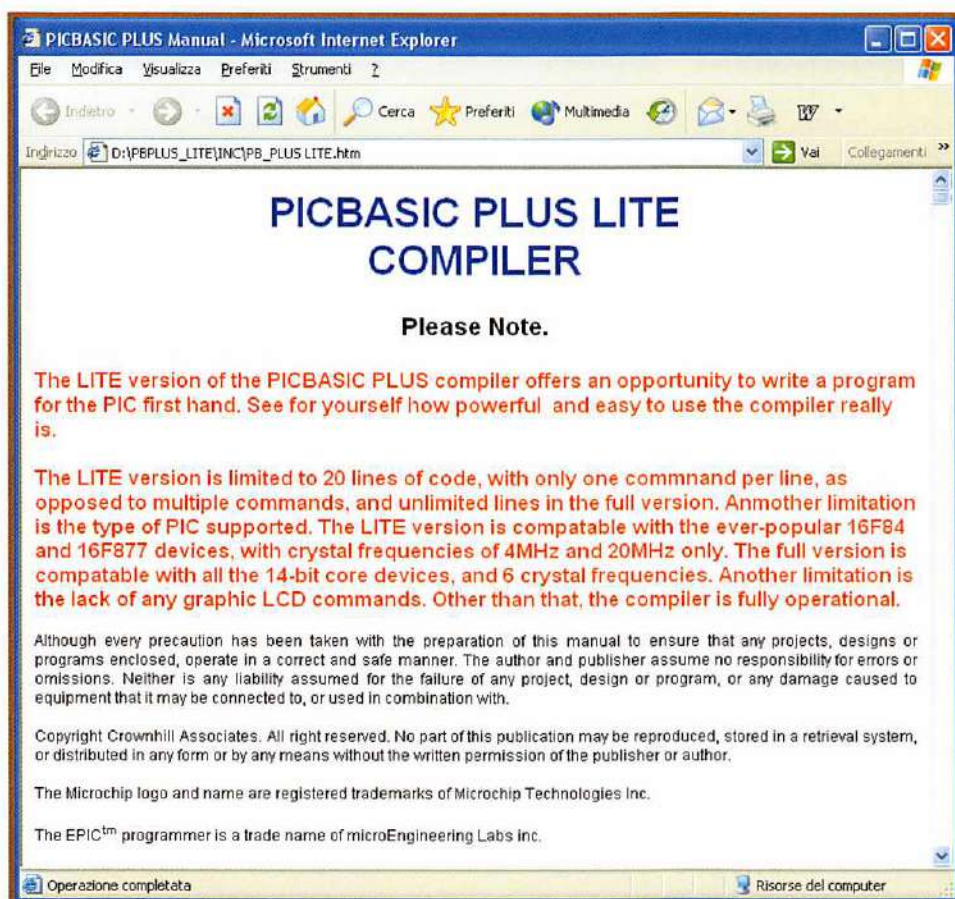
## Conclusioni

Il programma PICBASIC PLUS LITE è un esempio di come programmare un PIC con un linguaggio di programmazione di alto livello. Esistono compilatori per altri linguaggi, ma il

Basic è il più diffuso dopo l'assembler stesso. Presenteremo alcuni esercizi che servono da esempio per il funzionamento del programma, comunque dovreste essere voi stessi ad approfondire il linguaggio di programmazione. Nella cartella Programmi del primo CD allegato all'opera, oltre al

programma LetPicBasic Plus troverete anche una guida o manuale chiamato "picbasic plus lite manual" che vi servirà come aiuto per imparare a programmare in linguaggio Basic sulla piattaforma presentata.

Questo software è un altro utile e pratico strumento e ne verrà continuata la presentazione nei prossimi fascicoli.



Manuale di aiuto del PicBasic Plus Lite.





# Programmazione con il PICBASIC PLUS LITE

**P**er iniziare a programmare con il PicBasic Plus Lite dobbiamo analizzare la struttura generale che ha il codice realizzato con questo programma e conoscere le parole chiave. Presto potrete realizzare i vostri primi programmi in questo ambiente di programmazione.

## Ordine del programma

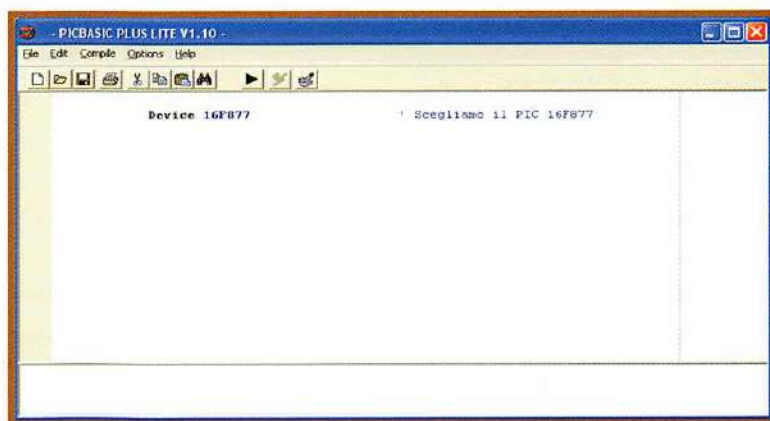
Dobbiamo abituarci a mantenere lo stesso ordine nella costruzione di un codice. Così come avviene nella programmazione in assembler tutte le varie parti o categorie di programmazione devono mantenere un particolare ordine. Questo sarà definito dal compilatore che ha un suo senso logico all'interno delle tecniche di programmazione. Tutti i programmi devono seguire la struttura riportata nella tabella sottostante.

La prima cosa da fare è definire il PIC

DEVICE	{tipo PIC}
DECLARE	{dispositivo} {valore}
DIM	{variabili}
SYMBOL	{nome} = {porta.pin}
DEFINE	{porta} = {input/output}
DATA	{tabelle} {...istruzioni di programma}
END	

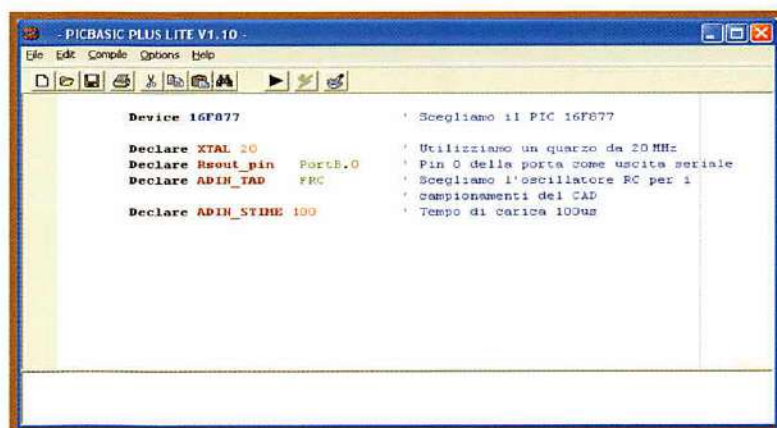
Struttura di un programma in PicBasic Plus Lite.

che vogliamo utilizzare. È necessario definirlo all'inizio del codice in modo che il compilatore possa interpretare l'informazione successiva all'interno del programma. Grazie all'istruzione "DEVICE" possiamo scegliere tra i modelli 16F84 e 16F877, dato che questa versione del software accetta solo PIC con architettura da 14 bit ed è limitata solamente a questi due modelli. Potremo applicare tutto ciò che abbiamo imparato per il PIC16F870, dato che il PIC16F877 è della stessa famiglia e riunisce tutte le caratteristiche che sono state spiegate per il PIC16F870.



Definizione del dispositivo tramite l'istruzione "DEVICE".

Dopo aver dichiarato il PIC che utilizzeremo possiamo definire i dispositivi esterni che verranno utilizzati nel programma. Quindi, mediante l'istruzione "DECLARE" assegniamo le variabili che sono necessarie per utilizzare dispositivi quali il display LCD, una tastiera, il bus I2C, la memoria dei dati EEPROM, la comunicazione seriale e i convertitori CAD. Sempre mediante questa istruzione dovremo defi-



Definizione degli elementi esterni "DECLARE".



```

- PICBASIC PLUS LITE V1.10 -
File Edit Compile Options Help
[Icons]

Device 16F877           ' Scegliamo il PIC 16F877

Declare XTAL 20        ' Utilizziamo un quarzo da 20 MHz
'Dichiarazione di variabili
DIM Topo as BYTE      ' Crea una variabile da 8 bit (0-255)
DIM Gatto as BIT      ' Crea una variabile da 1 bit (0-1)
DIM Cane as WORD      ' Crea una variabile da 16 bit (0-65535)

'Dichiarazione di alias
DIM Boxer as Cane.LOWBYTE ' Boxer è il primo byte (low byte)
                           della word Cane
DIM Terrier as Cane.HIGHBYTE ' Terrier è il secondo byte (high byte)
                           della word Cane
DIM Mickie as Topo.0   ' Mickie è il bit-0 di Topo
DIM CAT as Gatto

'Dichiarazione di costante
DIM Divisore as 16    ' Divisore avrà il valore 16
    
```

Esempio di utilizzo dell'istruzione "DIM".

nire la frequenza di funzionamento, ovvero il quarzo che utilizzeremo all'interno del circuito elettronico. Questa versione del software, oltre alla limitazione delle 20 linee di codice, ha una limitazione riguardo alle frequenze, dato che permette di lavorare solamente con due frequenze: 4 e 20 MHz. Nella figura possiamo vedere un esempio di come utilizzare questa istruzione in un programma che utilizza un convertitore analogico-digitale (CAD).

**Parole riservate  
che non potranno essere utilizzate  
come nomi riservati**

PP0	PP2H	PP5	PP7H	GEN4
PP0H	PP3	PP5H	GEN	GEN4H
PP1	PP3H	PP6	GENH	GPR
PP1H	PP4	PP6H	GEN2	BPF
PP2	PP4H	PP7	GEN2H	

Parole riservate.

```

- PICBASIC PLUS LITE V1.10 -
File Edit Compile Options Help
[Icons]

Device 16F877           ' Scegliamo il PIC 16F877

Declare XTAL 20        ' Utilizziamo un quarzo da 20 MHz
'Dichiarazione di variabili
DIM Topo as BYTE      ' Crea una variabile da 8 bit (0-255)

SYMBOL LED = PORTA.1   ' LED riferito al bit-1 della PortA
SYMBOL LED1 = PORTA.2 ' LED1 riferito al bit-2 della PortA
SYMBOL TOIF = INTCON.2 ' TOIF riferito al bit-2 del registro INTCON

SYMBOL Persiano = Gatto ' Gatto deve essere stato definito in precedenza con DIM

SYMBOL Topo = 1        ' Equivale a scrivere DIM Topo as 1
    
```

Diversi modi di utilizzare l'istruzione "SYMBOL".



```
- PICBASIC PLUS LITE V1.10 - D:\PBPLUS_LITE\file per EDVed65.bas
File Edit Compile Options Help
[Icons]
Device 16F877          * Scegliamo il PIC 16F877
Declare XTAL 20       * Utilizziamo un quarzo da 20MHz
DIM Gatto as BYTE     * Crea una variabile da 8 bit (0-255)
SYMBOL LED = PORTA.1  * LED riferito al bit-1 della PortA

Define TrisB.0 = 1    * bit-0 della PortB come ingresso
TrisB.1 = 0          * bit-1 della PortB come uscita
Trisa = %00000111    * bit2:0 della PortA come ingressi
Trisc = %10111111    * PortC.6 come uscita e il resto come ingressi
```

Definiamo i pin delle porte come I/O.

Con l'istruzione "DIM" si dichiarano le variabili del programma. Queste possono essere solamente bit, byte (8 bit) o word (16 bit). È obbligatorio dichiarare tutte le variabili prima di iniziare il loro utilizzo, dato che in caso contrario il compilatore ci darà un errore. Questa istruzione serve anche per delle variabili utilizzando dei nomi diversi, cioè potremo fare riferimento alla variabile utilizzando il suo nome oppure utilizzando un nome di comodo. Esistono una serie di parole riservate che non potranno essere utilizzate come nomi di variabili, riportate nella tabella della pagina precedente.

In ultimo, l'istruzione "DIM" può anche essere utilizzata per creare delle costanti, specificando il nome di quest'ultima e il valore fisso che avrà nel programma. Nella figura della pagina precedente è riportato un esempio di utilizzo fisso di istruzione per i diversi casi presentati. L'istruzione "SYMBOL" permette di dare un nome simbolico ai pin delle porte di ingresso/uscita. Questo nome normalmente viene assegnato in base all'utilizzo che si vorrà fare di questi pin e contribuisce a rendere più semplice la programmazione. Il nome può essere uno qualsiasi, la porta

deve essere A, B o C, e il pin è il numero di piedino all'interno della stessa porta. Ogni riferimento successivo a questo pin all'interno del programma potrà essere fatto utilizzando unicamente il nome che gli abbiamo assegnato.

Questa istruzione può anche essere utilizzata per dare un nome diverso a una variabile o a una costante, ma non può essere utilizzata per creare una variabile. Se creiamo una costante con questa istruzione non si potrà utilizzare la memoria RAM per contenerla.

Nella figura possiamo vedere diversi esempi dell'utilizzo dell'istruzione "SYMBOL".

"DEFINE" si utilizza per definire tutti i pin delle porte come ingresso o uscita. Di solito questa istruzione non si utilizza, dato che il compilatore permette di fare direttamente l'assegnazione. Ricordate che per definire i pin delle porte come ingressi o uscite, dovremo configurare i registri TRISX associati alle porte. Con il valore 1 configureremo un pin come ingresso e con il valore 0 come uscita.

Mediante l'istruzione "DATA" possiamo definire una tabella di dati alfanumerici, cioè una serie di caratteri alfabetici e/o numerici. Ognuno dei caratteri definiti è un dato a cui si può fare accesso per leggerlo. I caratteri alfabetici devono sempre essere contenuti fra virgolette

```
- PICBASIC PLUS LITE V1.10 - D:\PBPLUS_LITE\file per EDVed65.bas
File Edit Compile Options Help
[Icons]
Device 16F877          * Scegliamo il PIC 16F877
Declare XTAL 20       * Utilizziamo un quarzo da 20MHz
DIM Gatto as BYTE     * Crea una variabile da 8 bit (0-255)
SYMBOL LED = PORTA.1  * LED riferito al bit-1 della PortA

Define TrisB.0 = 1    * bit-0 della PortB come ingresso
TrisB.1 = 0          * bit-1 della PortB come uscita
Data "a","b","c",1,3,"Maria" * Creiamo una tabella di dati
```

Creiamo una tabella con l'istruzione "DATA".



(""), diversamente il compilatore interpreterà questo carattere come il nome di una variabile. Non tutte le istruzioni che abbiamo analizzato sono utilizzate insieme nello stesso programma, si utilizzano solamente quelle che sono strettamente necessarie. Mantenendo l'ordine logico nella programmazione, dopo aver preparato il "materiale" per lavorare, continuiamo con l'inserimento delle istruzioni del programma. Con queste istruzioni diciamo al PIC ciò che vogliamo fargli fare, le istruzioni che si utilizzano nel PicBasic Plus Lite, si possono trovare all'interno dell'help del programma stesso, nel documento in formato ".pdf" che si trova nella stessa cartella del primo CD, in cui è contenuto il programma.

Per dichiarare di aver terminato il codice dobbiamo inserire l'istruzione "END".

In questo modo il compilatore capirà che il codice è terminato e non compilerà nessun'altra istruzione.

## Etichette

L'utente può definire ciò che conosciamo come "etichette" ed evitare così di fare riferimento a indirizzi di memoria, quando deve eseguire dei file. Le etichette devono essere all'inizio della linea, e devono essere seguite da due punti (:) e uno spazio.

## Commenti

Nelle figure presentate come esempi avete potuto verificare che la spiegazione di ciò che esegue ogni linea è inserita alla destra della linea stessa, preceduta da una virgoletta sem-

```

Device 16F877           ' Scegliamo il PIC 16F877
Declare XTAL 20        ' Utilizziamo un quarzo da 20MHz
DIM Gatto as BYTE     ' Crea una variabile da 8 bit (0-255)
SYMBOL LED = PORTA.1  ' LED riferito al bit-1 della PortA
Define TrisB.0 = 1    ' bit-0 della PortB come ingresso
TrisB.1 = 0           ' bit-1 della PortB come uscita
Data "a","b","c",1,3,"Maria" ' Creiamo una tabella di dati

Inizio: Istruzioni
      ...
      ...
      ...
end
  
```

*Possiamo utilizzare etichette all'interno del codice.*

```

Device 16F877           ' Scegliamo il PIC 16F877
Declare XTAL 20        ' Utilizziamo un quarzo da 20MHz
DIM Gatto as BYTE     ' Crea una variabile da 8 bit (0-255)
SYMBOL LED = PORTA.1  ' LED riferito al bit-1 della PortA
Define TrisB.0 = 1    ' bit-0 della PortB come ingresso
TrisB.1 = 0           ' bit-1 della PortB come uscita
Data "a","b","c",1,3,"Maria" ' Creiamo una tabella di dati

Inizio: REM Il programma inizia qui
      REM istruzioni
      REM istruzioni
end
  
```

*Un altro modo di inserire commenti, "REM".*

plice ('). Dobbiamo utilizzare i commenti per rendere più facile l'interpretazione del codice. Un altro modo di definire un commento è l'utilizzo dell'istruzione "REM". Tutto ciò che sulla stessa linea arriva dopo questa istruzione verrà interpretato dal compilatore come un commento.

## Tabulazioni

È consigliabile che le linee di programma a eccezione delle etichette, siano tabulate. In questo modo otterremo anche che il codice si possa comprendere con maggiore facilità.

Mantenendo sempre un ordine e gli stessi criteri durante la programmazione, questo lavoro risulterà molto più semplice e ottimizzerà i tempi di sviluppo di una applicazione.



# Primo programma con il PICBASIC PLUS LITE

**D**opo aver analizzato separatamente il software PicBasic Plus Lite e la struttura di programmazione di questo ambiente, li uniremo studiandoli attraverso i programmi di esempio che sono allegati al software stesso.

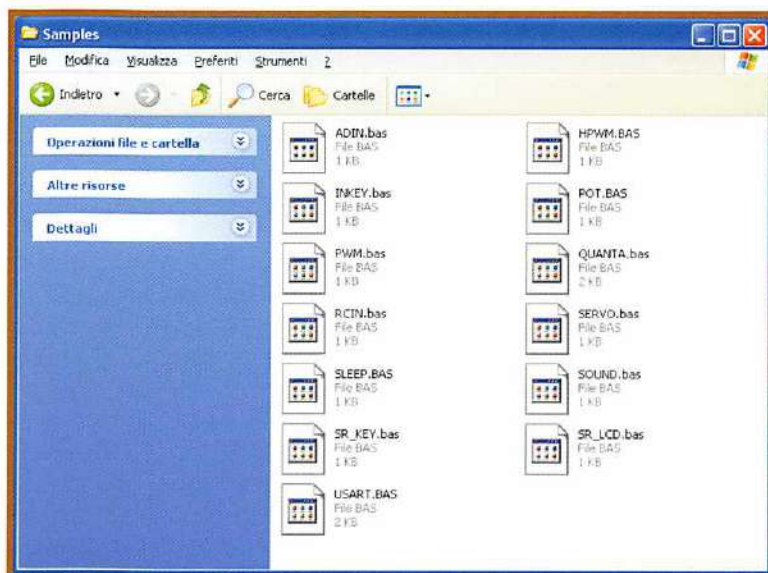
## Esercizi di esempio

All'interno della cartella che crea il programma PicBasic Plus Lite durante l'installazione, esiste una directory chiamata "Samples" dove potremo trovare alcuni programmi di esempio. I file di codice o programmi realizzati con questo software hanno come estensione ".bas". Quando salviamo un codice realizzato in questo ambiente di programmazione il file avrà l'estensione ".bas", ma se lo compiliamo viene creata una serie di file che abbiamo già visto altre volte: in assembler, in codice macchina, ecc.

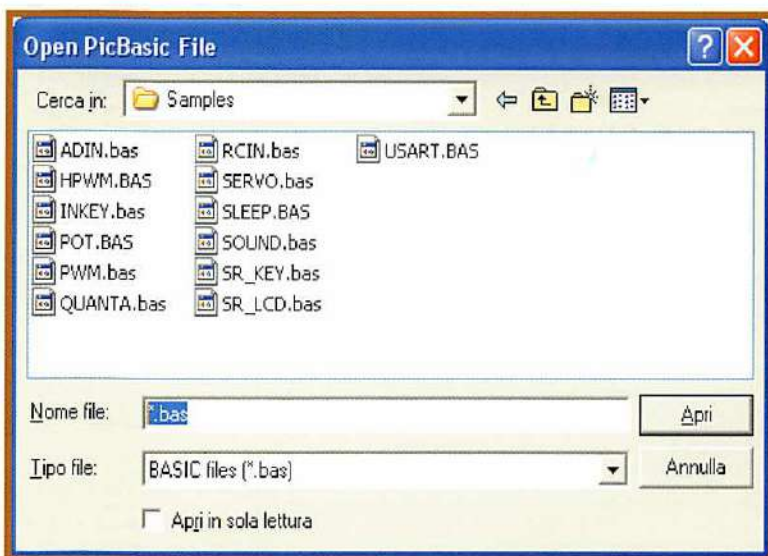
Nella figura possiamo verificare che i file si trovano nella cartella "Samples" e hanno nomi che ci permettono di capire a cosa sono riferiti senza doverli aprire e vederne il contenuto.

## Carichiamo un esempio sul PicBasic Plus Lite

Con il programma PicBasic Plus Lite aperto selezioniamo l'opzione Open nella barra degli strumenti o dal menù File. Facendo questo appare una finestra in cui è riportato il contenuto della cartella "Samples". Scegliete il primo dei file "ADIN.bas" e caricatelo. Adesso nell'editor è stato caricato il programma selezionato e possiamo vedere che nella prima linea viene spiegato, mediante un commento, ciò che esegue il programma. Questo programma legge il valore analogico che ha il pin 0 della porta A e lo visualizza.



Cartella "Samples", dove troveremo gli esercizi di esempio.



Selezionando l'opzione Open appare questa finestra.



```

- PICBASIC PLUS LITE V1.10 - C:\Programmi\BPPLUS_LITE\Samples\ADIN.bas
File Edit Compile Options Help
[Icons] [Run] [Stop] [Refresh] [Print] [Help]

' Read channel 0 of on-board ADC and display the results
Device 16F877
Declare XTAL 20
Declare Rcout_Pin PortB.0
Declare ADIN_TAD FRC
Declare ADIN_STIME 100
Dim Raw as Word
Dim Value as Word
Dim Volts as Byte
Dim Millivolts as Word
Rcout Cls
TrisA.0 = 1
ADCON1 = %10000000
Again: Raw = ADIN 0
Rcout at 2,1,"RAW= ",@Raw," "
Value = 489 *(Raw / 10)
Volts = Value / 10000
Millivolts=(Value // 10000) / 100
Rcout at 1,1,dec1 Volts,".",dec2 Millivolts,"V "
Goto Again
    
```

Carichiamo il programma "ADIN.bas".

verrà registrato il valore di ingresso, in Value convertiranno questo valore, in Volts scriveremo il valore in Volt e in Millivolts il valore in millivolt. Come vedremo, dichiarando le variabili si definisce anche il tipo o la dimensione delle stesse (bit, byte o word).

Create le variabili dovremo definire come ingressi o uscite i pin delle porte del PIC che utilizzeremo. Come abbiamo già visto, possiamo fare

Il programma ha una struttura familiare. Dopo il commento di inizio selezioniamo il PIC con cui vogliamo lavorare (16F877) e definiamo la frequenza con la quale funzionerà il microcontroller (20 MHz). Proseguiamo definendo i dispositivi esterni configurando il pin 0 della porta B come pin di uscita seriale verso l'LCD, l'oscillatore con cui lavorerà il convertitore analogo-digitale (RC) e impostando un tempo di carica al convertitore di 100 µs.

Continuiamo dichiarando le variabili che utilizzeremo all'interno del programma. In Raw

questo utilizzando l'istruzione DEFINE o direttamente assegnando un valore logico (0 o 1) al bit equivalente del registro associato TRISx. All'interno del programma che stiamo considerando viene fatto direttamente mediante l'istruzione "TrisA.0=1". È necessario configurare anche il registro ADCON1 per fare in modo che questo pin, definito come ingresso, sia di tipo analogico. Il ciclo che deve acquisire il valore dell'ingresso e mandarlo all'LCD in modo che lo visualizzi, inizia a partire dall'etichetta "Again". Si assegna alla prima variabile la lettura del pin di ingresso, lo visualizziamo e vediamo le operazioni sul valore per convertirlo in volt e millivolt e visualizziamo i risultati. Infine, saltiamo all'etichetta e ripetiamo il ciclo.

questo utilizzando l'istruzione DEFINE o direttamente assegnando un valore logico (0 o 1) al bit equivalente del registro associato TRISx. All'interno del programma che stiamo considerando viene fatto direttamente mediante l'istruzione "TrisA.0=1". È necessario configurare anche il registro ADCON1 per fare in modo che questo pin, definito come ingresso, sia di tipo analogico. Il ciclo che deve acquisire il valore dell'ingresso e mandarlo all'LCD in modo che lo visualizzi, inizia a partire dall'etichetta "Again". Si assegna alla prima variabile la lettura del pin di ingresso, lo visualizziamo e vediamo le operazioni sul valore per convertirlo in volt e millivolt e visualizziamo i risultati. Infine, saltiamo all'etichetta e ripetiamo il ciclo.

## Istruzioni

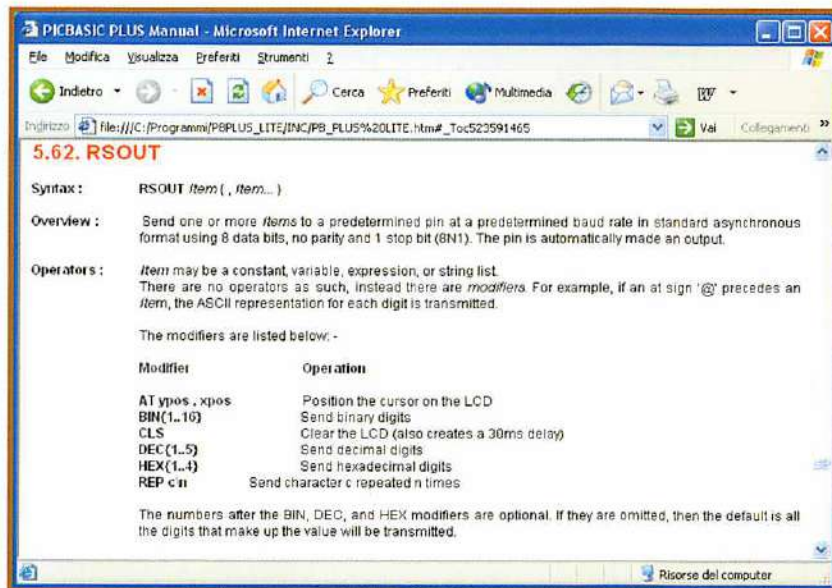
I programmi in Basic fanno in modo molto semplice ciò che

```

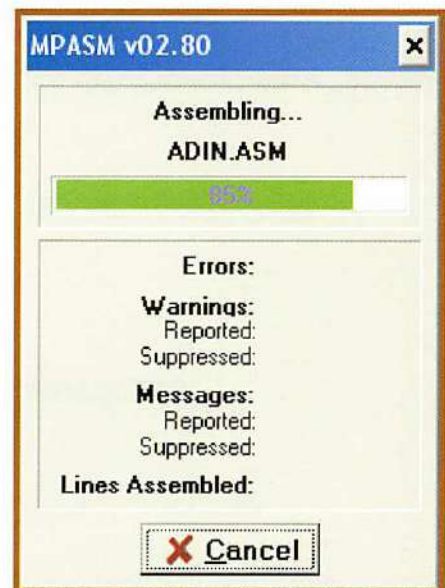
- PICBASIC PLUS LITE V1.10 - C:\Programmi\BPPLUS_LITE
Assembler Listing
AGAIN: R0B
      Clrw
      F0C rd8ad
      Mov0wz RAW
      Mov0wz PP7H
      Mov0wz RAWH
      B@fz BPFH,0
      Movlw 192
      F0C curs
      Movlw 'R'
      F0C Rcout
      Movlw 'A'
      F0C Rcout
      Movlw 'U'
      F0C Rcout
      Movlw ' '
      F0C Rcout
      Movlw 'V'
      F0C Rcout
      Movlw ' '
      F0C Rcout
      Mov0wz RAWH
      Mov0wz PP2H
      Mov0wz RAW
      Mov0wz PP2
      F0C o8dec2
      Movlw ' '
      F0C Rcout
      F0C Rcout
      F0C Rcout
      Clr0f BPFH
      Mov0wz RAMH
      Mov0wz PFOH
      Mov0wz RAW
    
```

PICBASIC PLUS COMPILED OK. 331 Words used  
33 Variables used in the 16F877 from a possible 368

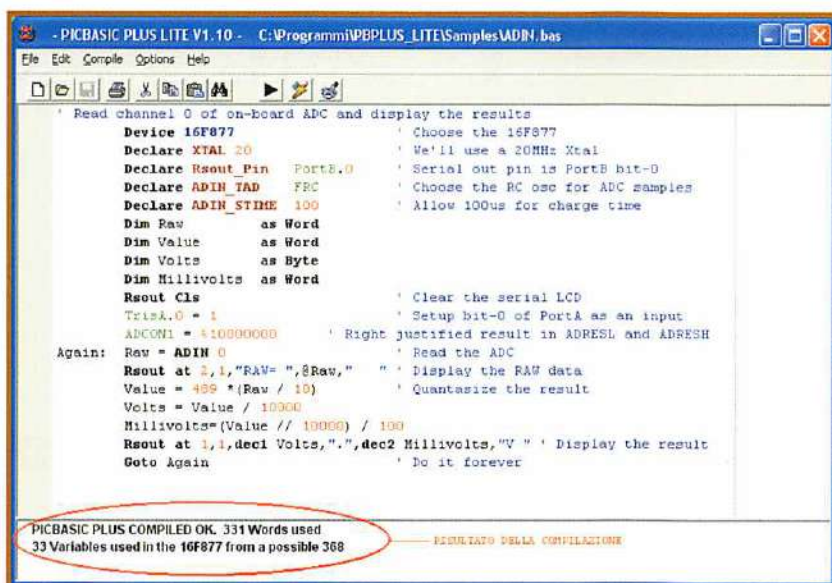
	Bit	Byte	Word
	1 bit	8 bit	16 bit
Raw		X	X
Value		X	
Volts			X
Millivolts			X



Aiuto del PicBasic Plus Lite.



Finestra di stato della compilazione.



Il risultato della compilazione è riportato all'interno dell'editor di testo.

in assembler può risultare complicato. Un'istruzione in Basic esegue più operazioni di un'istruzione in assembler.

Osservate ad esempio che in un'istruzione mandiamo il dato sul pin di uscita, inserendo un testo fisso e la variabile da visualizzare, mentre per fare la stessa cosa in assembler avremmo avuto bisogno di molte istruzioni. Nella figura è riportata la conversione eseguita dal programma stesso del file in assembler.

Nel manuale di aiuto del programma possiamo vedere come lavorare con le istruzioni di questo ambiente di programmazione, qui viene spiegata l'istruzione e il suo funzionamento e sono riportati esempi di utilizzo. Osservate nella figura il manuale di aiuto nella parte riferita all'istruzione RSOUT.

## Compilazione

Per poter vedere la conversione in assembler o in esadecimale (codice macchina) è necessario compilare il programma per fare in modo che generi questi file. Per compilare il programma dobbiamo cliccare sull'apposito pulsante che si trova nella barra degli

strumenti o selezionare l'opzione Compile Basic del menù Compile. In questo modo appare una finestra come quella nella figura in cui è riportato lo stato del processo di compilazione. Quando quest'ultimo si conclude, nella parte inferiore della finestra dell'editor o di lavoro appaiono una serie di informazioni relative ai risultati della compilazione e della dimensione del programma.

Se la compilazione è stata eseguita con successo vengono generati i file da essa risultanti,



```

Assembler Listing
*****
;*      Code Produced by the
;*      PICBASIC PLUS LITE Compiler, Version 1.10
*****

#include "ADIN.pbp"

RAW      Equ 59
RAVH     Equ 60
VALUE    Equ 61
VALUEH   Equ 62
VOLTS    Equ 63
MILLIVOLTS Equ 64
MILLIVOLTSH Equ 65

B=8f BPFH,6
FBC Cls
Clr#F BPFH
#Define TRISA_0 = 1
MovLw 128
Mov#of ADCON1
AGAIN
R8B
Clrw
FBC rd8ad
Mov#of RAW
Mov#of PP7H
Mov#of RAVH
B=8f BPFH,6
MovLw 192
FBC curs
MovLw 'R'
FBC Rcout
MovLw 'A'
FBC Rcout
    
```

Finestra del codice in Assembler e in linguaggio macchina.

quindi potremo selezionare nel menù Options la rappresentazione del codice in assembler (ASM) e/o codice macchina (HEX).

### Errori più comuni

Nel programma di esempio che abbiamo presentato ovviamente non esiste alcun errore, ma che cosa accade in presenza di qualche errore all'interno del codice? Come possiamo trovare gli errori?

Inseriamo degli errori all'interno del codice per analizzare ciò che succede. Nella figura abbiamo tolto una lettera all'interno del nome di un registro. Notate che la parola cambia di colore, dato che il compilatore cessa di riconoscerla come "speciale". Compilando il programma appare un messaggio in cui il compilatore ci informa che ci sono diversi errori di compilazione (Too many COMPILER ERRORS), però non dice la posizione dell'errore. Se l'errore lo inseriamo nella definizione di una va-

```

PICBASIC PLUS LITE V1.10 - C:\Programmi\PBPPLUS_LITE\Samples\ADIN.bas
File Edit Compile Options Help
*****
;*      Read channel 0 of on-board ADC and display the results
*****
Device 16F877
Declare XTAL 20
Declare Rcout_Pin PortB.0
Declare ADIN_IAD FRC
Declare ADIN_STIME 100
Dim Raw as Word
Dim Value as Word
Dim Volts as Byte
Dim Millivolts as Word
Rcout Cls
TRISA_0 = 1
ADCON1 = %10000000
Again: Raw = ADIN_0
Rcout at 2,1,"RAW= ",8Raw,"
Value = 499 *(Raw / 10)
Volts = Value / 10000
Millivolts=(Value // 10000) / 100
Rcout at 1,1,dec1 Volts,".",dec2 Millivolts,"V "
Goto Again
    
```

Inseriamo un errore nel nome del registro.

riabile, così come possiamo vedere nella figura a fianco, al momento della compilazione il messaggio ottenuto è diverso. In questo caso il compilatore ci dice il numero della linea in cui è stato rilevato l'errore e che tipo di errore ha trovato. Nella linea 18 manca l'assegnazione del risultato all'operazione, ovvero non è stata definita correttamente la variabile e quindi il risultato dell'operazione si assegna a "Millivolts" e questo nome non è riconosciuto dal compilatore. Provate altri possibili errori per familiarizzare con il programma e con i messaggi del compilatore. Caricate gli altri programmi di esempio ed esercitatevi con essi prima di iniziare a utilizzare questo software per applicazioni più complesse.

```

PICBASIC PLUS LITE V1.10 - C:\Programmi\PBPPLUS_LITE\Samples\ADIN.bas
File Edit Compile Options Help
*****
;*      Read channel 0 of on-board ADC and display the results
*****
Device 16F877
Declare XTAL 20
Declare Rcout_Pin PortB.0
Declare ADIN_IAD FRC
Declare ADIN_STIME 100
Dim Raw as Word
Dim Value as Word
Dim Volts as Byte
Dim Millivolts as Wrd
Rcout Cls
TRISA_0 = 1
ADCON1 = %10000000
Again: Raw = ADIN_0
Rcout at 2,1,"RAW= ",8Raw,"
Value = 499 *(Raw / 10)
Volts = Value / 10000
Millivolts=(Value // 10000) / 100
Rcout at 1,1,dec1 Volts,".",dec2 Millivolts,"V "
Goto Again
    
```

Inseriamo un errore nella definizione di una variabile.





## Esercizio: visualizzazione di messaggio, la pratica

**A**nche se non abbiamo ancora il display LCD, analizzeremo ora ciò che dovremo fare per provare sul laboratorio gli esercizi per la gestione del display LCD. Gli esercizi a cui fa riferimento questo montaggio sono "Ciao.asm" e "Addio.asm".

### Il display LCD

Con il prossimo fascicolo riceverete il display LCD e potrete quindi eseguire il montaggio descritto di seguito.

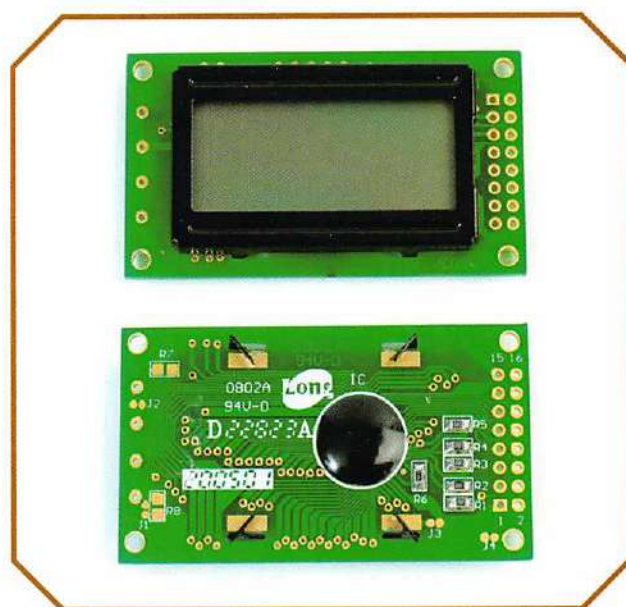
### Collegamento delle linee dei dati

Le 11 linee d'ingresso/uscita del modulo LCD, si differenziano in linee di dati e linee di controllo dell'LCD. Per trasferire qualsiasi comando o carattere utilizzeremo le linee dei dati D7:D0. Queste otto linee sono collegate ai due connettori volanti da quattro vie ciascuno, situati sul pannello superiore del laboratorio, sotto il display LCD. Come potete vedere nell'immagine della pagina successiva queste linee si collegano alla porta B, che è quella che assegneremo a tutte le applicazioni che lavorano con il display LCD alle linee dei dati.

È molto importante non incrociare i fili di collegamento, perché così facendo manderemo un dato sbagliato al display e non otterremo il risultato desiderato.

### Collegamento delle linee di controllo

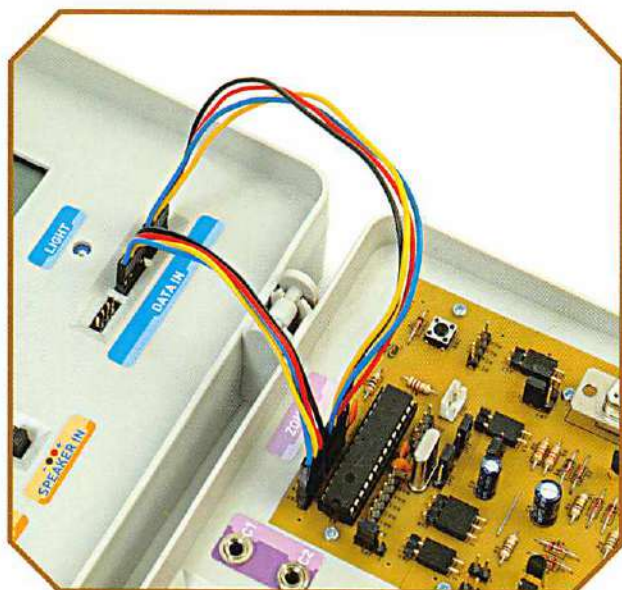
Per facilitare i collegamenti anche le tre linee di controllo E, R/W e RS sono accessibili dal pannello superiore del laboratorio. Il connettore volante a quattro vie situato alla sinistra della fila, corrisponde alle linee di controllo. Per fare in modo che il PIC controlli il display si assegnano come uscite i pin RA2:RA0 della porta A e si collegano alle linee di controllo. Utilizziamo un cavetto a quattro fili per effettuare il collegamento, anche se di questi ne useremo soltanto tre. Nelle immagini successive potete vedere come viene eseguito il collegamento di queste linee e che aspetto dovrà



Display LCD.



Il display LCD montato nel laboratorio.



Collegamento delle linee dei dati alla porta B del PIC.

assumere il pannello superiore del laboratorio dopo avere eseguito tutti i collegamenti.

I collegamenti spiegati in precedenza sono uno standard di collegamento, dato che li utilizzeremo ogni volta che lavoreremo con il display LCD. Riserveremo la porta B per le linee dei dati e i tre pin della porta A, RA2:RA0, per le linee di controllo.

## Regolazione del contrasto

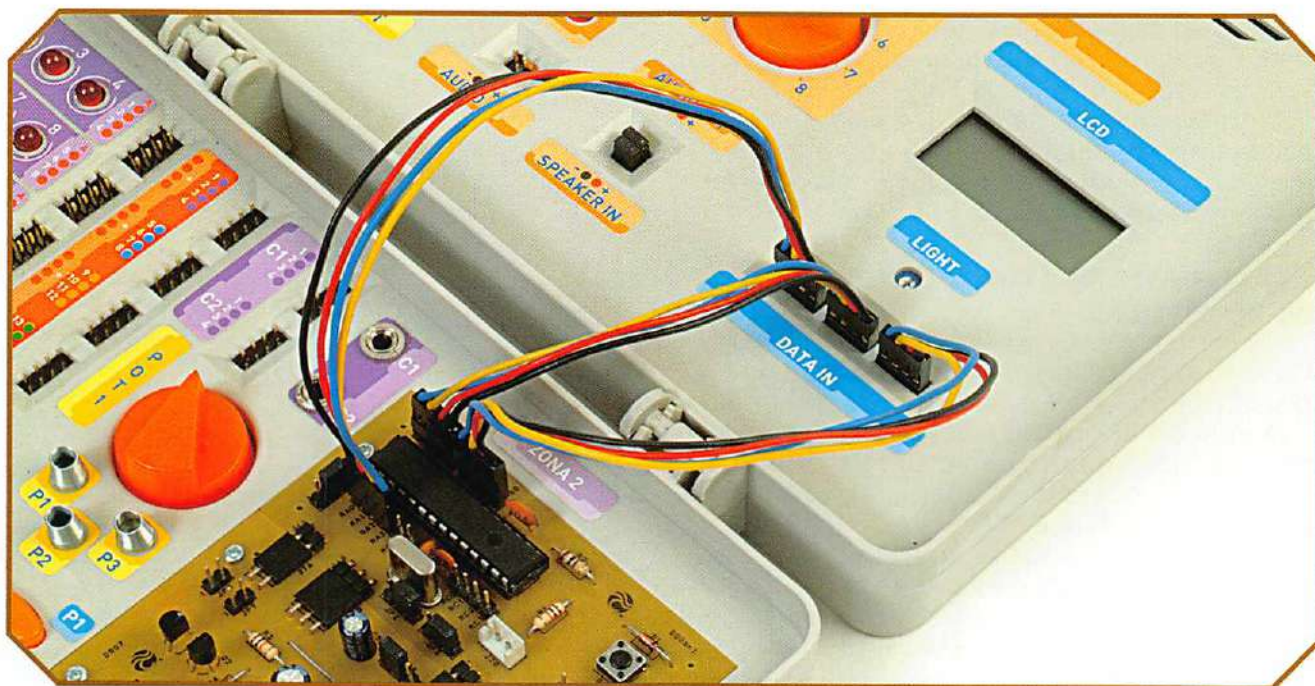
Mediante un potenziometro a resistenza variabile, collegato al modulo LCD, regoleremo il contrasto del display per poter vedere i messaggi visualizzati in esso. Dobbiamo regolare un potenziometro con un cacciavite a taglio sottile, sino a ottenere un'adeguata visione del carattere.

## Resto del montaggio

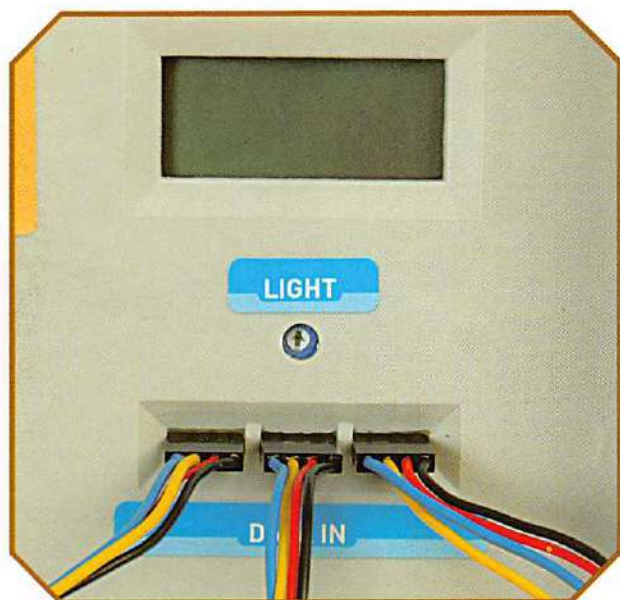
Sia nel programma "ciao.asm" che in "addio.asm" i messaggi si visualizzano direttamente sul display senza che vi sia nessuna condizione di ingresso. Quindi anche il codice di entrambi i programmi contempla il pin RA3 e RA4 come ingressi, ma questi non verranno utilizzati. Possiamo dire che il montaggio di questi due esercizi è concluso, perché sarà sufficiente caricare il programma sul microcontroller per verificarne il corretto funzionamento.

## Scrittura dei programmi sul PIC

Per scrivere i programmi sul microcontroller utilizzeremo il software IC-Prog, abbiamo bisogno però che il laboratorio sia configurato per poter realizzare il trasferimento corretta-



Collegamento delle linee di controllo.



Aspetto frontale del pannello superiore del laboratorio.

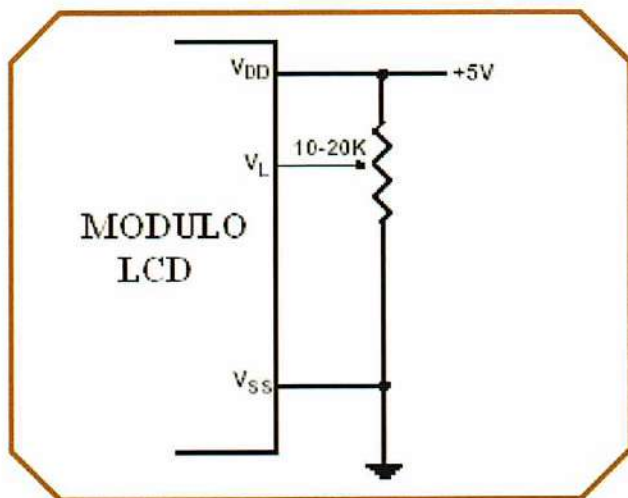


Regolazione del contrasto.

mente. Il microcontroller deve essere inserito nella posizione giusta, e i ponticelli della scheda DG06: JP1, JP2 e JP3, devono essere inseriti nelle posizioni 1 e 2. I ponticelli JP8 e JP9 devono anch'essi essere inseriti per poter scrivere il PIC, scollegeremo il connettore J63 delle porte da RB4 a RB7 in modo che non interferisca nella scrittura.

Infine, dobbiamo anche collegare il cavo di scrittura fra il laboratorio e il PC.

Facciamo partire IC-Prog e verificiamo che sia selezionato il PIC16F870 e che la scrittura venga eseguita tramite la porta selezionata.

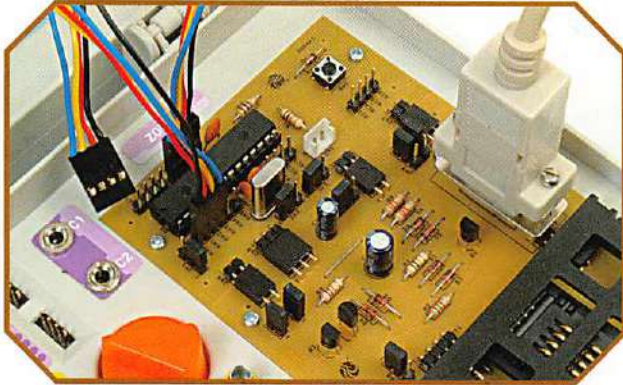


Schema di collegamento del potenziometro di contrasto.

Selezioniamo l'opzione Leggi tutto dal menù Comandi e premiamo F8 o selezioniamo il pulsante corrispondente dalla barra degli strumenti. Se sul PIC era già scritto un programma, vedremo cambiare i valori della finestra "Codice di Programma", quindi se lo riterremo opportuno, potremo salvare il nostro programma letto sul nostro PC.

Ora cancelliamo il dispositivo e a questo scopo selezioneremo l'opzione Cancella tutto. Essendo un processo critico ci sarà una finestra di conferma a cui risponderemo Yes, dando inizio alla cancellazione del dispositivo. Per verificare il processo leggeremo nuovamente il dispositivo, e verificheremo che su tutti gli indirizzi di memoria della finestra del codice del programma appaia il valore 3FFF.

Dobbiamo selezionare il programma che vogliamo scrivere sul PIC. Apriamo il primo file "ciao.hex" e controlliamo come cambia la finestra che contiene il codice del programma. Prima di scrivere configuriamo l'oscillatore selezionando XT, e la parola di configurazione con un segno di spunta su WDT e PWRT. Ricordate che è necessario mantenere l'opzione di protezione del codice su CP OFF. Selezionando Programma tutto, caricheremo il programma scelto sul PIC. Anche nel caso in cui si ottenga un messaggio di verifica corretta è consigliabile rileggere nuovamente il dispositivo, per verificare direttamente che la procedura sia stata eseguita con successo.

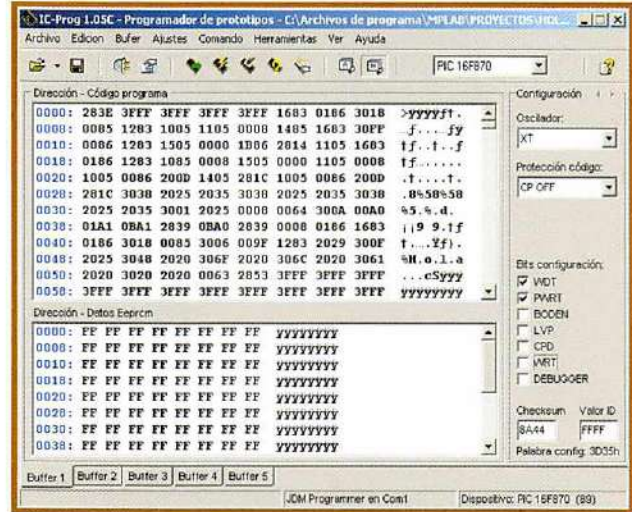


Configurazione del laboratorio per scrivere il PIC.

Ora abbiamo scritto il programma desiderato. Per caricare un altro programma dobbiamo ripetere i passaggi visti fino a questo momento.

### Prova di funzionamento

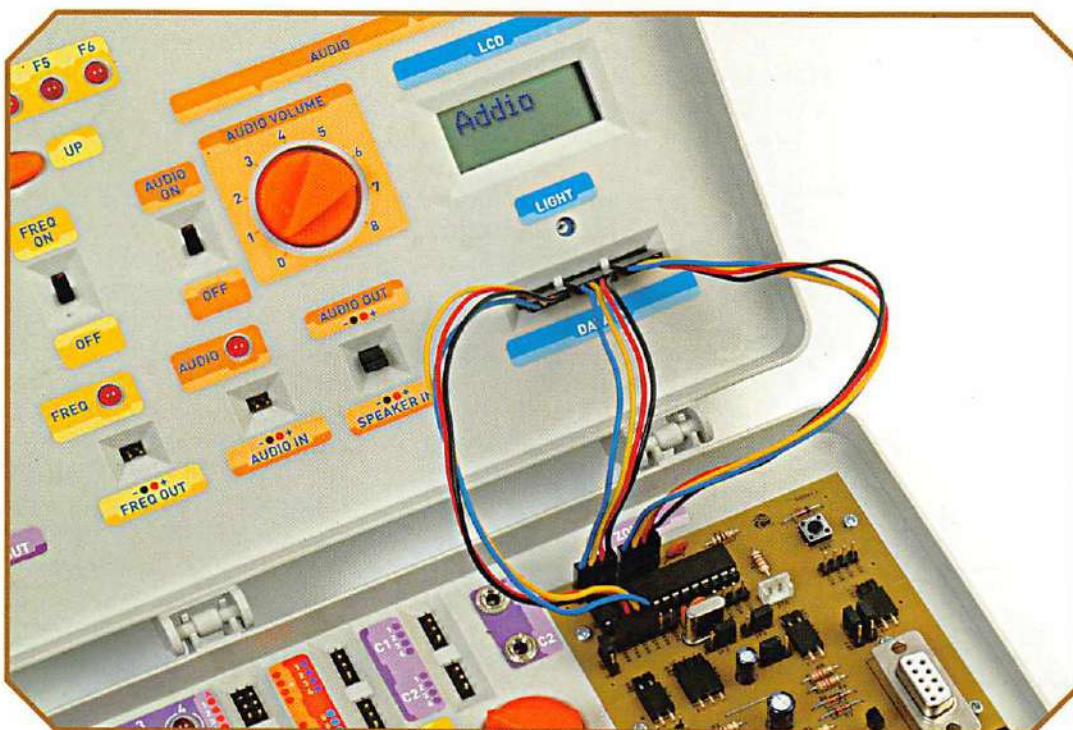
Dopo aver caricato il primo dei programmi dobbiamo riportare il laboratorio nella configurazione precedente alla scrittura. Scollegiamo il cavo di collegamento tra il laboratorio e il PC, collochiamo i ponticelli dei connettori JP1, JP2 e JP3 sui pin 2 e 3, e togliamo i ponticelli dei connettori JP8 e JP9. Facendo



Caricamento del primo programma su IC-PROG.

questo e alimentando il circuito sull'LCD si presenterà il messaggio "Ciao".

Se ripetiamo gli stessi passaggi per il programma "Addio.hex" verificheremo che i messaggi "Ciao" e "Addio" si alterneranno ogni 2 secondi. Nel caso in cui abbiate eseguito le modifiche che vi abbiamo consigliato per quest'ultimo programma, verificate che i diversi messaggi si alternino nello stesso modo di quelli appena descritti.



Aspetto del laboratorio durante la prova di funzionamento.



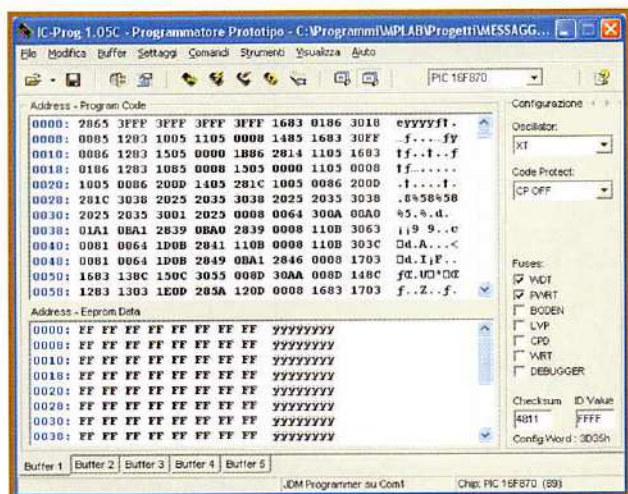
# Esercizio: generatore di messaggi, la pratica

**C**oncludiamo l'esercizio del generatore di messaggi scrivendo il programma sul microcontroller e realizzando il montaggio hardware dell'applicazione. Benché le applicazioni che comprendono il display LCD richiedano più programmazione o una programmazione più complessa rispetto ad altri tipi di applicazioni, i montaggi dei circuiti risultano piuttosto semplici.

## Scrittura del PIC

Dopo aver simulato il programma e verificato che non abbia errori, lo scriveremo sul PIC. Ricordate che per la scrittura è necessario predisporre il laboratorio, a questo scopo collegheremo il cavo di trasferimento e inseriremo i ponticelli sui connettori JP8 e JP9 e sui connettori JP1, JP2 e JP3 sulle posizioni 1 e 2. Non dimenticate che i terminali della porta B devono essere scollegati, in quanto potrebbero generare errori nel processo di scrittura.

Facciamo partire IC-Prog ed eseguiamo le fasi di preparazione alla scrittura (lettura, cancellazione e verifica). Apriamo il file che vogliamo trasferire sul PIC, "messaggio.asm", e configuriamo IC-Prog in modo che il trasferimento venga eseguito correttamente. Per questo cambieremo il tipo di oscillatore a XT, verificheremo che la protezione del codice sia CP OFF e selezioneremo WDT e PWRT tra i bit di configurazione. A questo punto daremo il via alla scrittura selezionando l'opzione Programma Tutto. Terminata la fase di scrittura



Carichiamo il programma su IC-Prog.

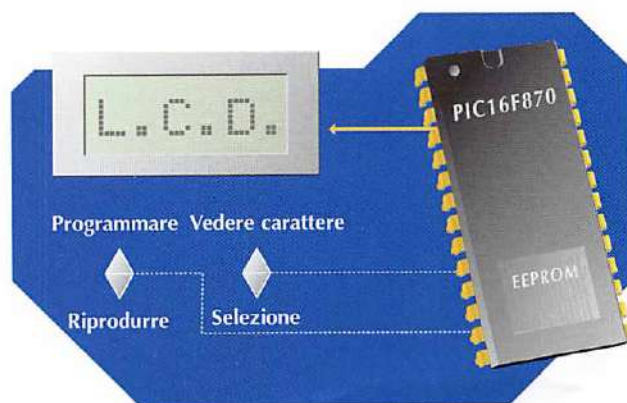
verificheremo il processo eseguendo una lettura del contenuto del PIC.

## Montaggio hardware

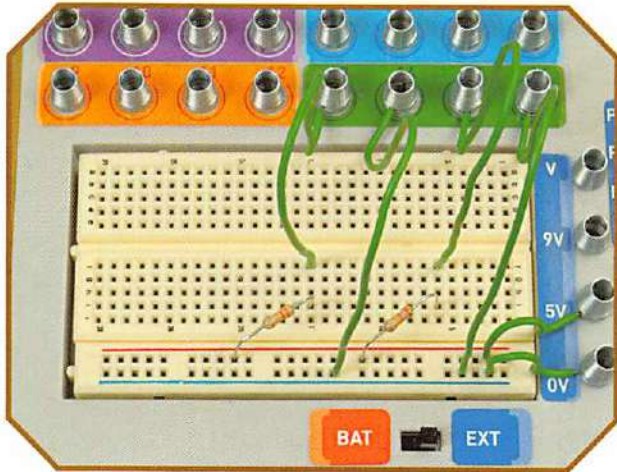
Facendo riferimento allo schema generale di funzionamento separeremo il montaggio in due parti: montaggio degli ingressi (RA3 e RA4) e montaggio delle uscite (RA2:RA0 e la porta B). Prima di procedere ai montaggi dobbiamo riportare il laboratorio in modo funzionamento, quindi scollegheremo il cavo di trasferimento dei dati tra il PC e il laboratorio, toglieremo i ponticelli da JP8 e JP9 e sposteremo i ponticelli di JP1, JP2 e JP3 dalla loro posizione, portandoli sui pin 2 e 3.

## Montaggio degli ingressi

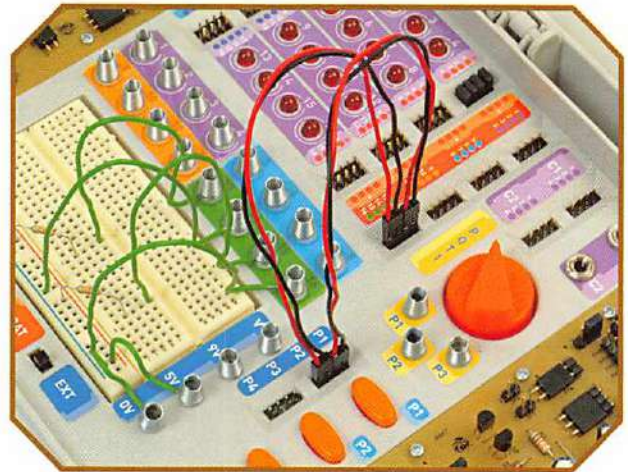
Abbiamo bisogno di due ingressi, uno per selezionare programmare/riprodurre e l'altro per selezione/vedi caratteri, i quali all'interno del codice sono stati assegnati a RA4 e RA3 rispettivamente. Nella scheda Bread Board del laboratorio inseriremo due resistenze che attraverso un lato si collegheranno al positivo (5 V) mentre l'altro capo si col-



Schema generale del funzionamento.



Montaggio degli ingressi sulla scheda Bread Board.

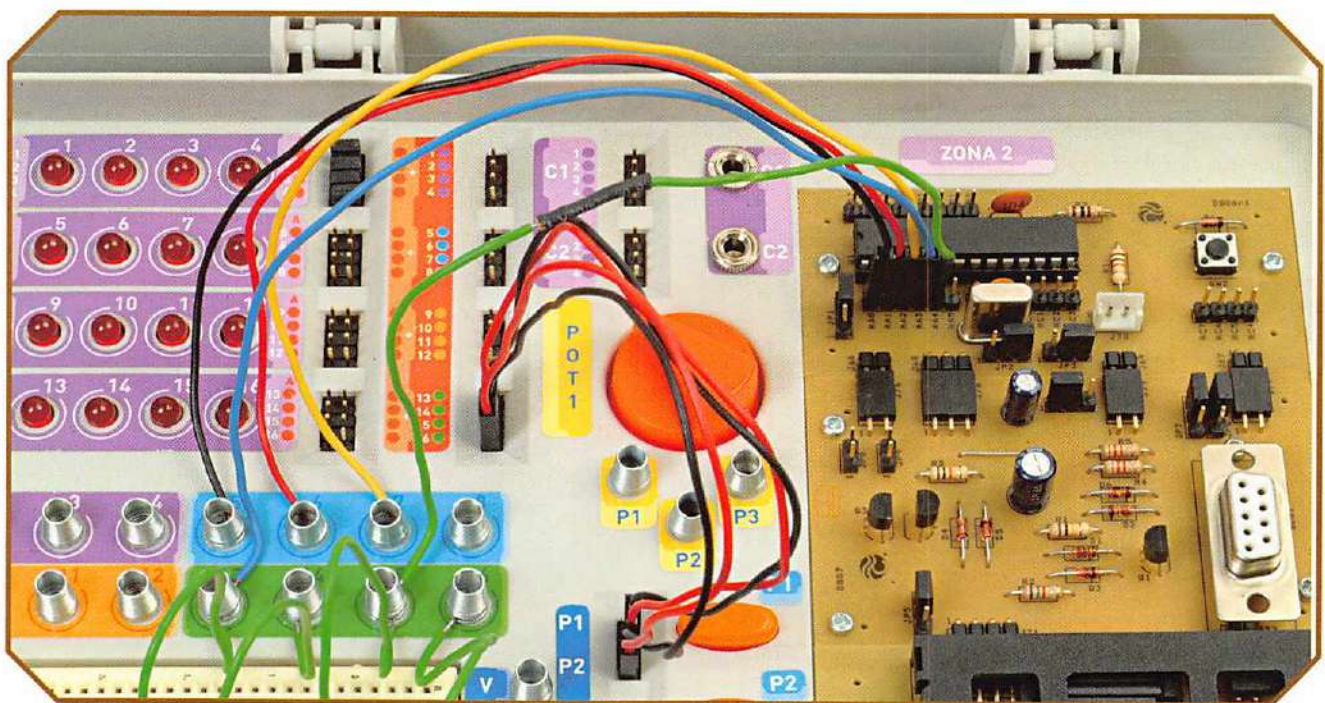


Collegamento dei pulsanti per il montaggio degli ingressi.

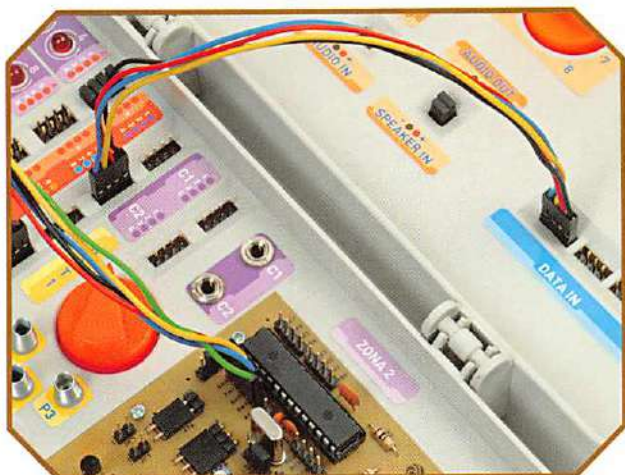
legherà alle molle 13 e 15. Le molle 14 e 16 sono unite mediante dei fili al negativo (0). Nella figura potete vedere il montaggio che stiamo spiegando.

Ora dobbiamo montare i pulsanti al montaggio precedente, a questo scopo utilizzeremo due cavetti a due fili per unire i connettori corrispondenti alle molle di collegamento verdi con i connettori di P1 e P2. Il pulsante P1 si collegherà alle molle 13 e 14 e il pulsante P2

alle molle 15 e 16, come potete vedere nell'immagine in alto. Per portare gli ingressi al PIC è sufficiente unire la molla 13 con RA3, che corrisponde a P1 e la molla 15 a RA4, che corrisponde a P2. Per effettuare questo collegamento utilizzeremo un filo diretto dalla molla 15 al pin RA4 e uno dei fili di un connettore da quattro per collegare RA3 alla molla 13. Gli altri tre fili del cavetto di collegamento verranno portati alle molle azzurre 5, 6 e 7, in modo



Collegamento degli ingressi sul PIC.



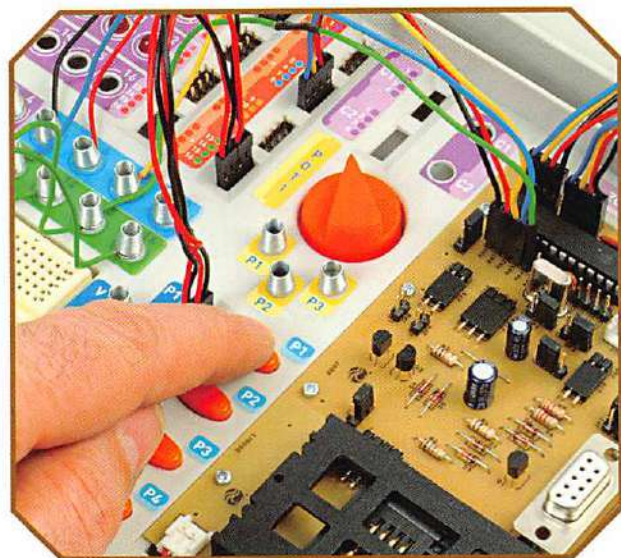
Portiamo all'LCD i segnali di E, RS e R/W.

che la molla 5 sia collegata con RA0, la 6 con RA1 e la 7 con RA2.

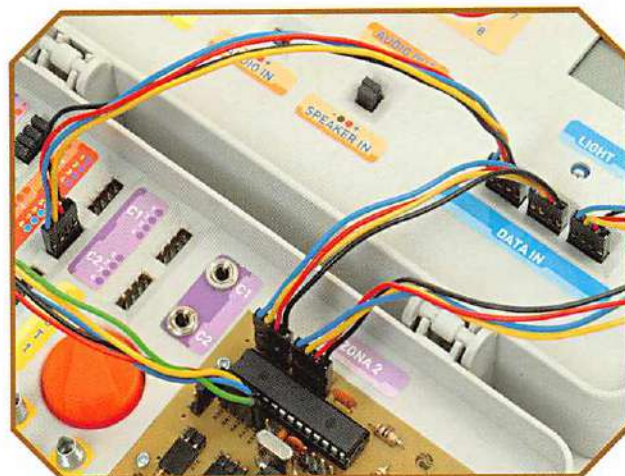
Per terminare il montaggio degli ingressi dobbiamo collegare il connettore corrispondente alle molle azzurre con il display LCD. Mediante un cavetto uniamo i due connettori, in modo che la molla 8 corrisponda al pin situato più a sinistra dei connettori del display LCD del pannello superiore.

## Montaggio delle uscite

Per eseguire il montaggio delle uscite è sufficiente collegare gli 8 pin della porta B con le 8 linee dei dati dell'LCD. Realizzeremo i col-



Premiamo P1 per scrivere un carattere.



Collegiamo le uscite del PIC all'LCD.

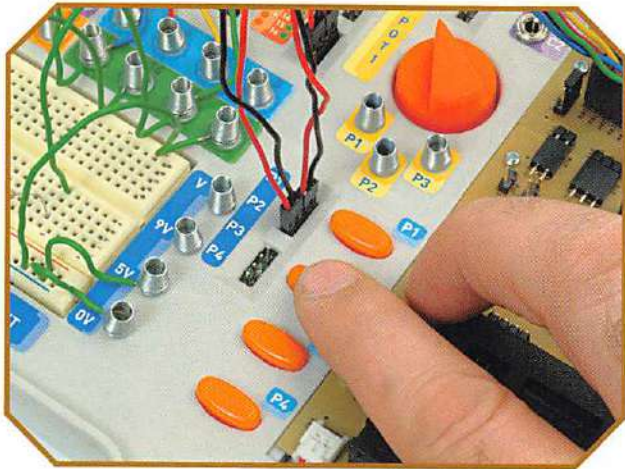
legamenti con due cavetti da quattro fili, tenendo presente che non possiamo incrociare i cavetti, perché se succedesse, i dati inviati al display sarebbero sbagliati. Nella figura potete vedere come è stato realizzato questo collegamento.

## Prova di funzionamento

Questo esercizio è uno dei più impegnativi da provare sul laboratorio. Acquisire la scioltezza necessaria per scrivere un messaggio correttamente, all'inizio richiederà un po' di allenamento, dato che si potrebbe incontrare qualche difficoltà a utilizzare i pulsanti.

Quando diamo alimentazione al laboratorio, agli ingressi RA4 e RA3 arrivano 5 V, dato che non sono premuti i pulsanti P1 e P2, il programma quindi si trova in modo programmazione, mostrando i caratteri in sequenza. I caratteri vengono visualizzati per il tempo definito nel programma. Per selezionare il primo carattere del messaggio dovremo premere P1 in modo che arrivi uno 0 sull'ingresso RA3. Quando il pulsante ritorna nella sua posizione originale il programma visualizza nuovamente tutti i caratteri sequenzialmente. In pratica dovremo premere P1 tante volte quante sono i caratteri contenuti nel messaggio.

Quando il messaggio sarà completato lo potremo riprodurre premendo P2. Con questa azione arriverà uno 0 sull'ingresso RA4, quindi il PIC entrerà in modo riproduzione e si potrà vedere il messaggio sul display LCD

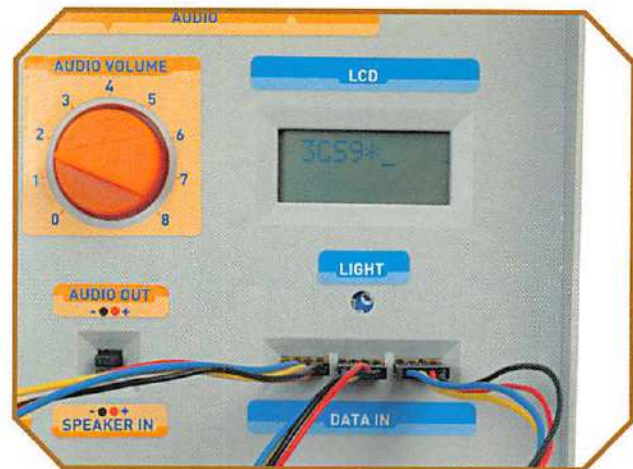


La riproduzione del messaggio si ottiene azionando P2.

per il tempo programmato. Nell'immagine potete vedere come viene visualizzato un messaggio temporaneo, precedentemente scritto.

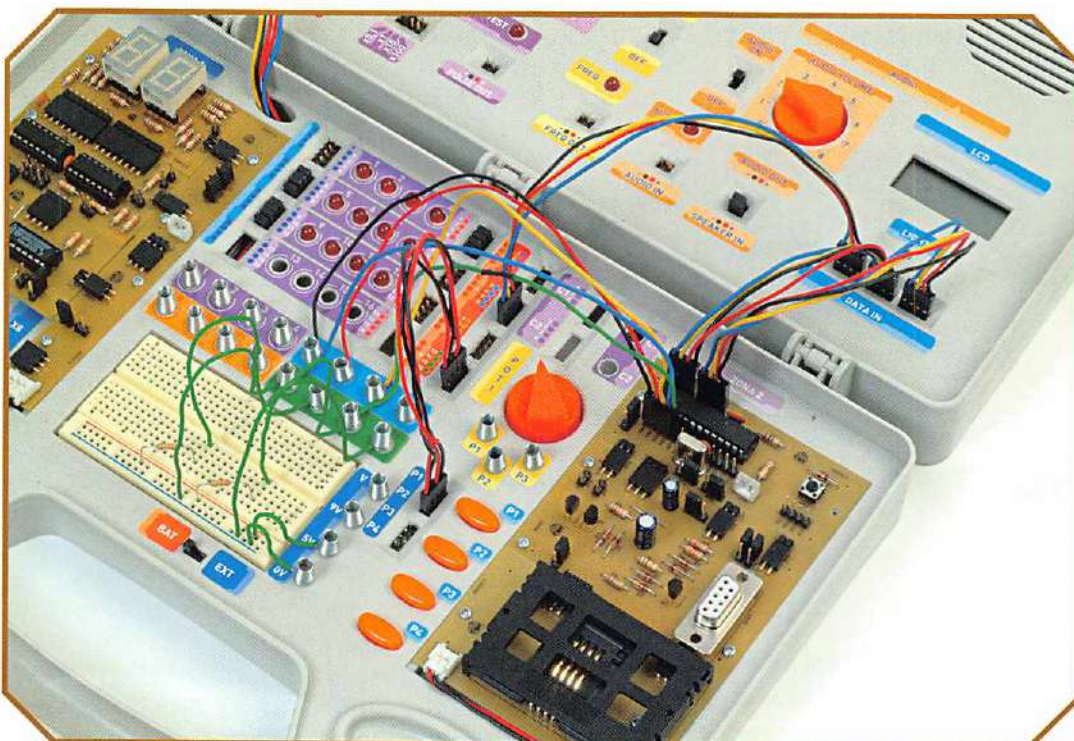
## Conclusioni

Abbiamo risolto questa applicazione in modo completo, ma ci sono ancora alcuni aspetti riguardanti il funzionamento del display LCD che non sono ancora stati evidenziati, ad



Riproduciamo il messaggio sul display LCD.

esempio cosa succede quando si scrive un messaggio di più di 8 caratteri, che cosa succede con la memoria EEPROM quando si scollega l'alimentazione dal laboratorio, che risultato otteniamo dalle diverse possibili combinazioni degli ingressi o come agiscono sull'LCD i segnali E, RS e R/W. Provate, esercitatevi e modificate i programmi se lo riterrete necessario, dato che tutto questo vi aiuterà ad affrontare nuove applicazioni con maggior confidenza ed esperienza.



Aspetto generale del laboratorio.