

Classe : **4A TIEN**
A.S. : 2005-2006
Docente : Prof. Franco Tufoni Codocente: Prof. Enrico Ruggieri
Disciplina : Sistemi, automazione e organizzazione della produzione

Nutchip

Teoria ed Esercizi

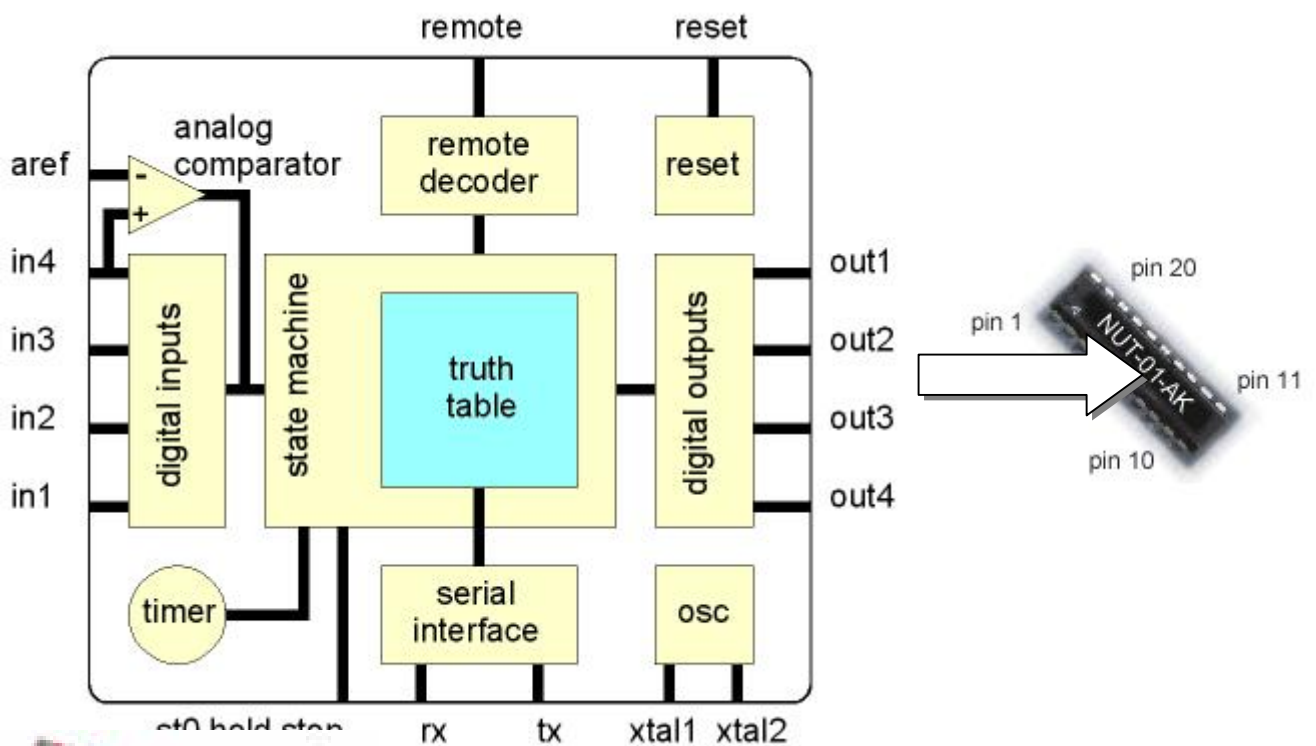


Fig. 1



Vers. 4.0

Introduzione

I Nutchip sono circuiti integrati programmabili che consentono di realizzare in modo rapido e semplice piccoli automatismi (*automi a stati finiti*) di media complessità.

Un Nutchip e qualche componente esterno è tutto ciò che serve per realizzare migliaia di circuiti diversi: un antifurto, una fotocellula, un timer, un dado elettronico, un metronomo, un generatore di impulsi, un termostato, comando motore passo passo, un campanello intelligente: sono solo alcuni esempi di ciò che si può realizzare con i Nutchip.

Lo stesso circuito integrato può essere riprogrammato più volte per svolgere funzioni diverse.

A differenza di altri integrati, la programmazione dei Nutchip non richiede l'utilizzo di costosi dispositivi programmatori e nemmeno la conoscenza di specifici linguaggi di programmazione.

Per la programmazione si utilizza il software **Nutstation**, si compila una tabella che compare a video, per determinare le combinazioni di ingressi e uscite da ottenere. Quindi si trasferisce la matrice al chip tramite la porta seriale o USB del computer.

Altre caratteristiche interessanti dei Nutchip sono:

-) la possibilità di comandarne il funzionamento attraverso un *telecomando* a raggi infrarossi o a onde radio;
-) la presenza di un *timer* interno per generare temporizzazioni da 1 ms a 1000 ore.

Per comprendere il funzionamento e la programmazione dei Nutchip, sono necessarie alcune premesse.

Automi a stati finiti

Un automa a stati finiti è un sistema sequenziale, le cui uscite in un certo istante di tempo dipendono non soltanto dai valori degli ingressi in quell'istante, ma anche dalla particolare *configurazione* in cui il sistema si trova. In ciò si differenzia dai sistemi combinatori in cui le uscite dipendono unicamente dai valori presenti sugli ingressi.

Un semplice esempio può chiarire il concetto: si pensi a un sistema dotato di un solo ingresso, a cui è collegato un pulsante, e di una sola uscita, a cui è collegato un LED. Il suo funzionamento è il seguente:

ogniquale volta viene attivato il pulsante, il LED si spegne se era acceso e, viceversa, si accende se era spento.

L'attivazione del pulsante ha dunque effetti diversi a seconda che il sistema si trovi nella configurazione con il LED acceso o con il LED spento.

Le diverse configurazioni che un sistema può assumere nel suo funzionamento sono dette **stati** e il passaggio da uno stato all'altro viene detto **transizione**.

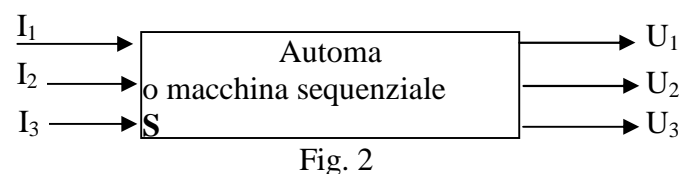
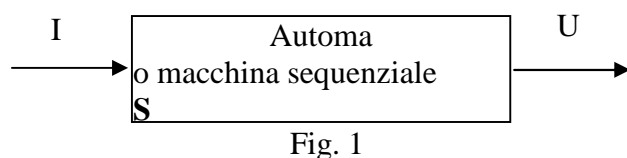
In un automa a stati finiti o sistema sequenziale l'uscita dipende dalla sequenza degli ingressi e dallo stato del sistema.

In modo sintetico tutto questo si esprime con i seguenti simboli matematici:

$$U=f(S,I)$$

$$U=Uscita; I=Ingresso; S=Stato del sistema$$

Il sistema può essere composto da un ingresso ed una uscita (Fig.1) oppure più ingressi e più uscite (Fig.2).



Rappresentazione formale degli automi a stati finiti

Per la rappresentazione degli automi a stati finiti (A.F.) oppure macchine sequenziali vengono usate forme grafiche (diagramma di transizione degli stati) oppure forme tabulate (tabella di transizione degli stati).

Lo stato in cui si trova il sistema prima della **transizione** è detto *stato presente* (stato n -esimo), mentre lo stato che il sistema raggiungerà dopo la transizione è detto *stato futuro* (stato $n+1$ -esimo). Da uno stato presente si possono verificare più transizioni che portano a stati futuri differenti.

Diagramma di transizione degli stati

In questo caso gli stati del sistema sono rappresentati graficamente da un cerchio (o ellisse), all'interno del quale si riporta il nome convenzionale dello stato (S_0, S_1, S_3, \dots) con i valori che le uscite assumono in quello stato.

Per indicare la transizione da uno stato all'altro, causata dagli ingressi, i cerchi corrispondenti agli stati interessati dalla transizione vengono raccordati con un arco orientato, contrassegnato dai valori degli ingressi che determinano la transizione stessa.

Esempio

Si consideri il funzionamento di un latch *Set-Reset* realizzato con porte NOR, il cui schema a blocchi è riportato in fig. 3.

Il sistema presenta due stati (S_0 e S_1). In S_0 le uscite Q e \bar{Q} assumono, rispettivamente, il valore 0 e 1. In S_1 le uscite assumono il valore $Q = 1$ e $\bar{Q} = 0$. Gli ingressi *Set* e *Reset* sono attivi alti e, per ipotesi si supponga che non possano attivarsi contemporaneamente (in tal caso si avrebbe la condizione non usata $Q=\bar{Q}=00$). La fig. 4 riporta il diagramma di transizione; degli stati che descrive il funzionamento del latch.

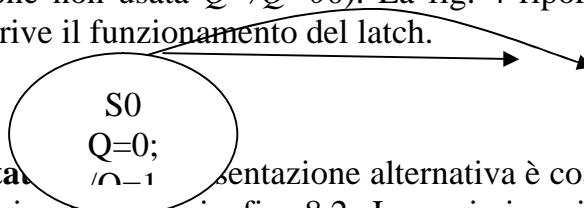


Tabella di transizione degli stati La rappresentazione alternativa è costituita dalla tabella di transizione degli stati organizzata come in fig. 8.2. In ogni riga si indicano: lo stato presente, il valore assunto dalle uscite in quello stato, le possibili combinazioni degli ingressi, gli stati futuri corrispondenti.

La tabella di transizione degli stati per il sistema dell'esempio precedente è riportata in fig. 8.2b. Se il sistema si trova nello stato S_1 , il verificarsi della condizione 1, 0 sugli ingressi SR porta il sistema nello stato futuro S_1 ossia il sistema rimane nello stato in cui si trovava prima. Con il sistema nello stato S_0 , il verificarsi della stessa condizione 1, 0 sugli ingressi SR provoca invece una transizione allo stato S^1 . Quindi, il verificarsi di una stessa condizione sugli ingressi causa un funzionamento del sistema diverso in dipendenza dello stato presente in cui esso si trova.

La tabella di transizione degli stati del Nutchip

Il Nutchip è un sistema a stati finiti il cui funzionamento è descritto mediante una tabella di transizione degli stati che viene memorizzata nella sua memoria interna.

Per ogni stato di funzionamento descritto nella tabella, il Nutchip (precisamente il suo blocco interno denominato macchina a stati, l'esecutore della tabella) attende il verificarsi di una fra le condizioni di ingresso specificate per quello stato. Al verificarsi di una condizione sugli ingressi, l'esecutore porta il sistema nello stato successivo indicato nella tabella.

Il Nutchip è dotato di quattro uscite digitali, $out1, out2, out3, out4$, e di quattro ingressi digitali $in1, in2, in3, in4$. I valori che devono assumere le uscite in un certo stato e le

condizioni logiche che devono verificarsi sugli ingressi per ottenere la transizione allo stato futuro desiderato sono assegnati definendo una speciale tavola di verità o di transizione. Questa costituisce sostanzialmente il programma.

Il Nutchip può inoltre gestire sei tasti di un eventuale telecomando attraverso *REMOTE*. I pulsanti del tei sono convenzionalmente *key1, key2...key6* e sono da tutti gli effetti come altrettanti digitali. La tabella di fransi; Nutchip è organizzata come Interpretazione. La tabe sizione del Nutchip rii fig. 8.3 è del tutto simile alle tabelle di fig. 8.2 con alcune eccezioni relative a *REMOTE* e alla presenza dei *timeout*.

O Nella colonna *state* verranno riportati gli stati *presenti* del sistema. I nomi coi li da assegnare agli stati sono *stOQ, s* 01, stQ2...* Lo stato 5/00 è quello inizia Nutchip si porta non appena viene fornita tensione (*power-up*) o dopo un-res si stati che compongono la tavola sono separati da una linea orizzontale.

Nelle quattro caselle della colonna **output 1...4**, andranno specificati i valori 0 o 1 che devono assumere le uscite nello stato considerato (nell'ordine da sinistra verso destra: *OUT1, OUT2, OUT3 e OUT4*).

o Nella colonna **input 1...4** verranno specificate le condizioni sugli ingressi che determinano la transizione nello stato futuro (nell'ordine *INI, IN2, IN3 e 77V4*). Per uno stesso stato è possibile specificare più condizioni d'ingresso diverse. Per ciascuna condizione di ingresso occorre specificare nella colonna **next** lo stato futuro, ovvero la transizione di stato provocata da quella condizione.

e Quando non interessa specificare un valore per un ingresso, ovvero è indifferente se esso assumerà valore 0 o 1 (*dont'caré*), nella casella corrispondente dell'ingresso si inserirà un trattino «-», che ha lo stesso significato del simbolo X usato di solito nei manuali tecnici. In pratica, la presenza di un trattino in corrispondenza di un ingresso fa sì che esso non venga preso in considerazione nella valutazione della condizione. Con riferimento alla fig. 8.3, per esempio, nello stato 5/00 il verificarsi della condizione *INI = 1 e IN2 = 0* provoca la transizione allo stato 5/02, indipendentemente dai valori logici presenti sugli ingressi *IN3 e IN4*.

e La colonna **remote** si riferisce al telecomando; qui andranno inseriti i nomi assegnati ai pulsanti del telecomando: *key1, key2...key6*. La condizione si verifica quando il tasto premuto corrisponde a quello riportato nella casella della colonna **remote**. Inserendo tre trattini «—» il telecomando verrà ignorato. È anche possibile utilizzare il telecomando in combinazione con gli ingressi. Nella tabella di transizione di **fig. 8.4**, l'attivazione del tasto *key3* provoca la transizione allo stato 5/03 solo se, contemporaneamente, risulta verificata la condizione: *INI = 1, IN2 = 0 e IN4 ~ 0*; l'ingresso *IN3* non viene preso in considerazione. Se non interessa mettere in relazione gli ingressi con i tasti del telecomando, allora nella colonna **input 1...4** si indicherà la condizione di indifferenza («—») come in **fig. 8.5**.

o Una caratteristica interessante del Nutchip, che si rivela utile in numerose applicazioni, è la possibilità di inserire nella tabella di transizione *temporizzazioni* di durata programmabile (da 1 ms a 1000 ore) sotto forma di condizioni speciali dette **timeout**.

Nella condizione *timeout* occorre specificare la durata della temporizzazione e lo stato successivo in cui il sistema dovrà portarsi allo scadere dell'intervallo di tempo prefissato.

-La temporizzazione ha inizio non appena il sistema si porta nello stato in cui è presente il *timeout*. Allo scadere del tempo programmato, se nessun'altra condizione risulta verificata, il sistema si porta nello stato successivo indicato nella condizione *timeout*. La condizione *timeout* è valutata sempre per ultima rispetto alle altre eventuali condizioni presenti nello stato. Si consideri ancora la tabella di fig. 8.3: non appena il sistema si porta nello stato 5/01 viene avviata una temporizzazione della durata di 30 minuti. Allo scadere dei 30 minuti, il

sistema si porta nello stato 5/02 soltanto se il tasto *keyl* non risulta contemporaneamente attivo, altrimenti si porta nello stato

Durante l'esecuzione di un *timeout*, il verificarsi di una condizione sugli ingressi o dovuta al telecomando viene accettata bloccando la temporizzazione e reinizializzando il timer interno. Per esempio (fig. 8.6), se durante il *timeout* di 30 minuti viene premuto il tasto *keyl* del telecomando, il timer viene azzerato e il sistema si porta nello stato 5/00.

Descrizione del NUTCHIP

In tabella 1 sono riportate le caratteristiche tecniche, in tabella 2 il pin-out dell'integrato.

<i>Caratteristiche tecniche</i>			
▶ <i>nessun linguaggio di programmazione</i> , basta introdurre una tabella della verità			
▶ versatile multi-timer programmabile da 1 millesimo di secondo a ben 1000 ore!			
▶ decodificatore per telecomando incorporato, fino a 6 tasti programmabili a piacere. Riconosce sia i codici a radiofrequenza sia i codici a infrarossi			
▶ comparatore analogico incorporato per fissare soglie, collegare direttamente sensori, etc.			
▶ 3 ingressi digitali , oppure (il comparatore si può utilizzare come quarto ingresso digitale)			
▶ 4 uscite digitali ad alta corrente capaci di pilotare direttamente i LED (fino a 20 mA per ogni uscita)			
▶ 3 ingressi dedicati per gli automatismi meccanici: restart, hold, stop			
▶ porta seriale per collegamento al PC			
▶ programmabile più 10.000 volte in-circuit, cioè con il chip già montato nel circuito di destinazione, senza necessità di smontarlo nè cancellarlo			
▶ versatile base dei tempi: a quarzo per la massima precisione oppure con un economico oscillatore ceramico			
▶ alimentazione da 4 a 6 volt stabilizzata			
<i>Absolute maximum ratings</i>		<i>DC characteristics</i>	
Voltage on any pin except /RESET with respect to ground	-1.0 to +7.0V	Input low voltage (V)	-0,5...(0.2V _{cc} -0.1)
Storage temperature	-40°C to 105°C	Input high voltage (V)	0.2V _{cc} +0.9...V _{cc} +0.5
Maximum operating voltage	6.6 V	Internal pull-up resistor	35...100 kohm
Maximum output pin current	40 mA	Output current: source	10 mA
Maximum current V _{cc} to GND	140 mA	Output current: sink	20 mA
		Analog comparator offset	20 mV max
		Analog comparator leakage current	10 nA max
		Max clock frequency	10 MHz

Tabella 1

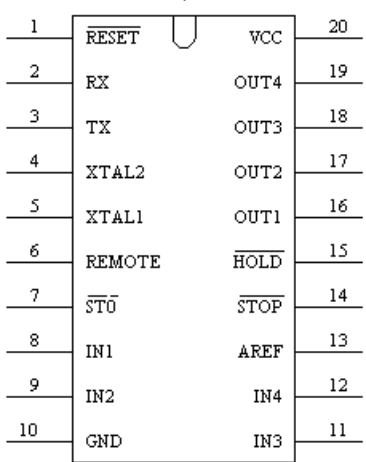

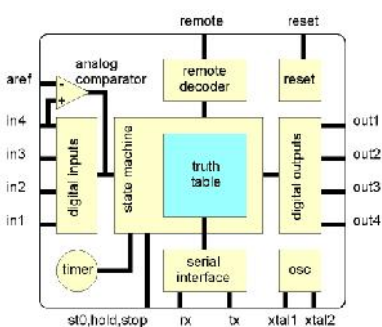
Nome pin	Descrizione	Pin	
/RESET	Reset del chip: mantenendolo basso per 2 cicli di clock il dispositivo si resetta	1	 <p><i>Pinout del Nutchip.</i></p>  <p><i>Contenitore dual-in-line (DIL)</i></p>  <p><i>Schema a blocchi interno</i></p>
RX, TX	Programmazione del chip (da collegare alla porta seriale del PC)	2, 3	
XTAL2	Uscita dell'oscillatore di clock	4	
XTAL1	Ingresso oscillatore di clock o ingresso per clock esterno (4MHz)	5	
REMOTE	Ingresso del decoder di telecomando	6	
/ST0	Ingresso di soft-reset: mantenendolo basso per 10mS il chip si porta nello stato ST0	7	
IN1...IN2	Ingressi digitali generici della macchina a stati	8, 9	
GND	Negativo di alimentazione (ground)	10	
IN3	Ingresso digitale generico della macchina a stati	11	
IN4 (COMP)	Ingresso digitale generico della macchina a stati, oppure ingresso positivo del comparatore analogico (l'ingresso vale 1 se $V_{in} > A_{ref}$)	12	
AREF	Ingresso tensione di riferimento del comparatore analogico	13	
/STOP	Ingresso di STOP: mantenendolo basso per 10mS l'esecuzione si arresta	14	
/HOLD	Ingresso di HOLD: mantenendolo basso per 10mS il timer viene disabilitato	15	
OUT1...OUT4	Uscite digitali programmabili dalla macchina a stati	16...19	
Vcc	Alimentazione 5V (da 4 a 6V)	20	

Tabella 2

Schema a blocchi del Nutchip

In fig 1 è riportato lo schema a blocchi dell'integrato.

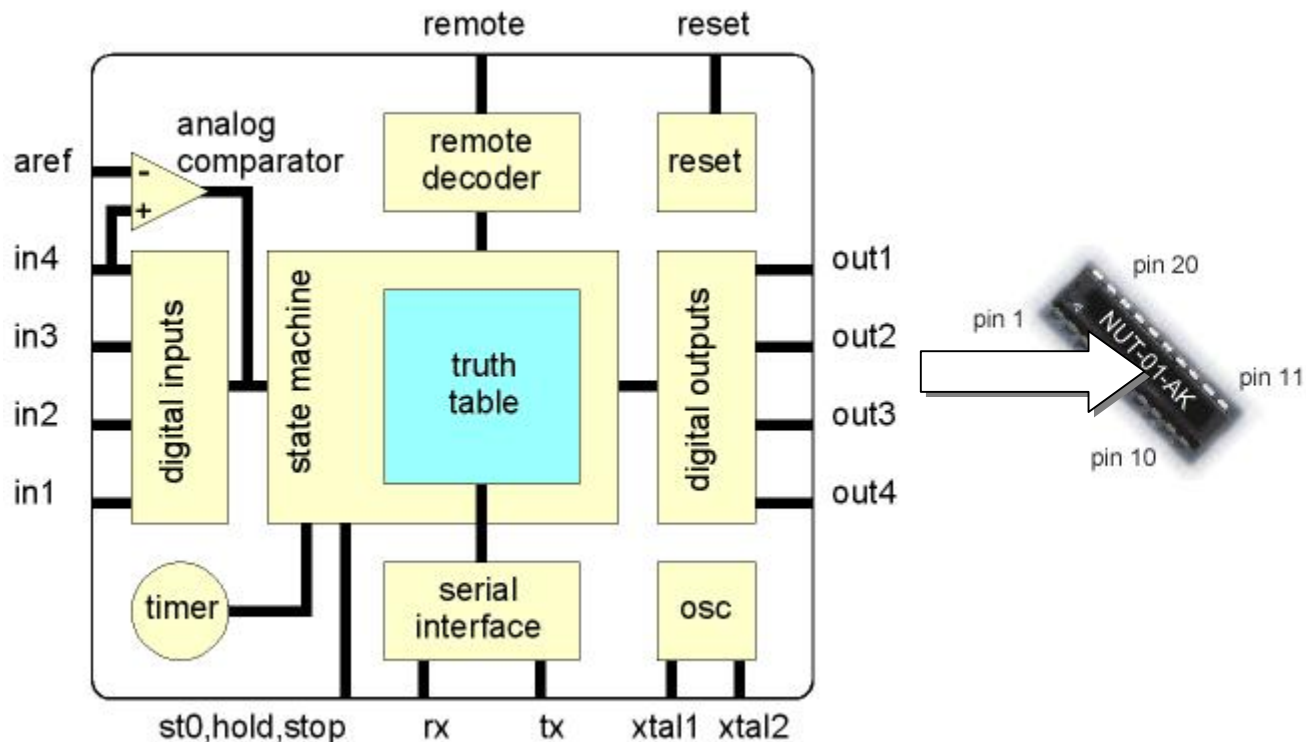


Fig. 1

State machine: elemento principale della struttura, il cui funzionamento è stabilito dalla tavola di verità (truth table) immagazzinata nella memoria interna non volatile del dispositivo. È possibile memorizzare tavole di verità fino a un massimo di 32 stati, 64 condizioni, 6 codici di telecomando e 30 timeout differenti; in realtà la capacità effettiva può variare a seconda del tipo di applicazione. La tavola di verità è programmabile da PC attraverso l'interfaccia seriale di cui è dotato il Nutchip e con l'utilizzo del software Nutstation.

Serial interface: il blocco consente il collegamento del Nutchip alla porta seriale del PC attraverso i pin RX e TX che andranno collegati, mediante opportuni traslatori di livello TTL-RS-232, rispettivamente ai pin TD e RD) della porta seriale del PC. Tale collegamento risulta necessario in fase di programmazione e per l'utilizzo del telecomando virtuale Nutchip Commander.

Remote decoder: il blocco è il circuito decodificatore del segnale ricevuto attraverso l'ingresso REMOTE da un ricevitore per telecomando. Infatti è possibile comandare il Nutchip con un telecomando a raggi infrarossi o a radiofrequenza. Il decoder può riconoscere sino a 6 codici, programmabili a piacere, corrispondenti a sei diversi tasti del telecomando. A ogni codice-tasto, il Nutchip associa le variabili booleane, convenzionalmente chiamate key1, key2...key6, che possono assumere i valori logici Vero/Falso. Per esempio, la variabile key2 è associata al pulsante 2 del telecomando. Premendo il pulsante 2, key2 risulterà Vera, altrimenti il suo valore sarà Falso. In pratica, lo stato Vero/Falso delle variabili rispecchia lo stato dei pulsanti, per cui nel seguito key1...key6 verranno chiamate indifferentemente pulsanti o variabili. Le variabili key1...key6 possono essere inserite nella tavola di verità e sono gestite come se fossero normali ingressi del Nutchip.

Timer: è il timer programmabile a 32 bit, che consente di effettuare temporizzazioni da 1

ms a 1000 ore.

Analog comparator: comparatore analogico, attraverso cui è possibile gestire tutti quei sensori che forniscono un segnale analogico anziché digitale, come per esempio fotoresistenze o fotocellule. Il comparatore è dotato di due ingressi analogici: AREF (analog reference) e IN4. Su questi ingressi è possibile applicare una tensione compresa fra 0 e 5V. Sull'ingresso AREF verrà applicata una tensione di riferimento, mentre sull'ingresso IN4 verrà applicata la tensione di uscita del sensore. L'uscita del comparatore è digitale e può assumere i livelli logici 0 o 1; essa avrà il valore logico 0 se la tensione presente su IN4 è minore della tensione presente sull'ingresso AREF, mentre assumerà il valore 1 se la tensione su IN4 risulta maggiore della tensione presente su AREF. Il comparatore può essere attivato oppure disattivato in fase di programmazione del Nutchip. Nel caso in cui il comparatore risulti disattivato, l'ingresso IN4 diventa un normale ingresso digitale.

Reset: il blocco consente di resettare il Nutchip. Mantenendo l'ingresso RESET a livello logico basso per 2 periodi di clock, il sistema viene azzerato e riportato nello stato iniziale st00.

Osc: blocco oscillatore, genera il segnale di clock ($f_{ck} = 4 \text{ MHz}$) necessario al funzionamento del Nutchip. Sui pin Xtal1 e Xtal2 occorre collegare un quarzo o un filtro ceramico da 4 MHz.

Digital inputs: ingressi programmabili. Il Nutchip è dotato di quattro ingressi digitali: IN1, IN2, IN3 e IN4. Internamente essi sono collegati a Vcc mediante resistori dipull-up (da $35 \text{ k}\Omega$ a $100 \text{ k}\Omega$); ciò consente, per esempio, di collegare interruttori agli ingressi senza utilizzare resistori di Pull-up esterni. Tali ingressi sono programmabili nel senso che i loro valori (0 o 1) intervengono nella tavola di verità per determinare la transizione dello stato. Come si è detto, l'ingresso IN4 normalmente funziona come ingresso digitale ma può essere programmato come ingresso analogico assegnandolo al comparatore.

Digital outputs: uscite programmabili. Il Nutchip è dotato di quattro uscite digitali OUT1, OUT2, OUT3 e OUT4 programmabili. Per assegnare un valore logico a una qualunque delle uscite basta scriverlo nella corrispondente casella della tavola di verità. Le uscite sono di tipo push-pull e quindi sono in grado sia di erogare corrente (corrente di source, massimo 10 mA, fig. 2) quando sono a livello alto, sia di assorbire corrente (corrente di sink, massimo 20 mA, fig. 3) quando sono a livello basso.

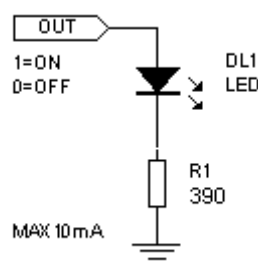


Fig. 2

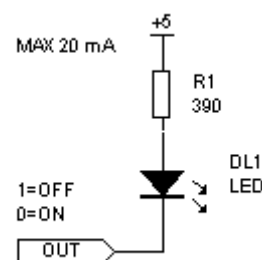


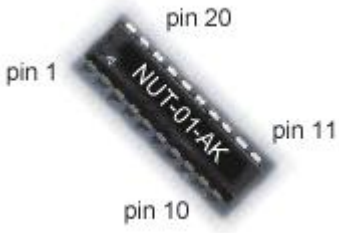
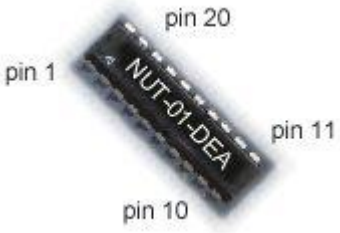


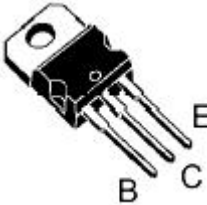
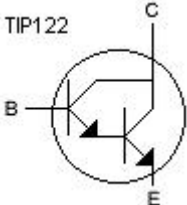
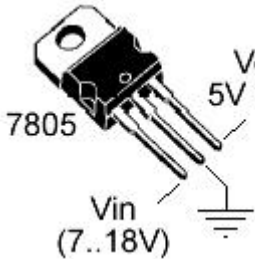
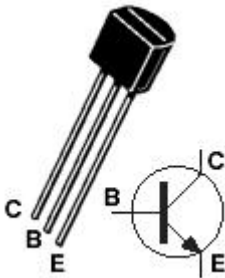
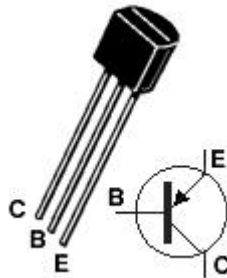

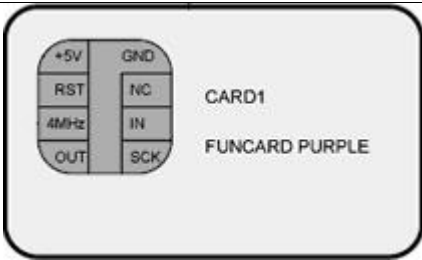
Fig. 3

Ingressi predefiniti: oltre agli ingressi programmabili, IN1, IN2, IN3 e IN4, il Nutchip è dotato di altri quattro ingressi che svolgono funzioni predefinite e indipendenti dalla tavola di verità. In tabella 3 è descritta la loro funzione. Gli ingressi predefiniti hanno la priorità sulle condizioni d'ingresso stabilite nella tavola di verità. Se più ingressi predefiniti risultano contemporaneamente attivi, l'ordine di esecuzione non è definito.

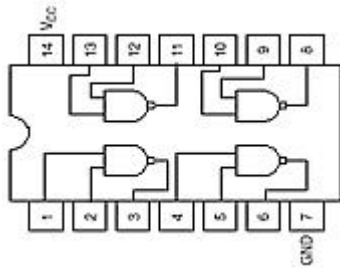
Ingresso	Pin	Effetto	Descrizione
/ST0	7	Riporta il funzionamento del Nutchip nello stato iniziale ST00	L'attivazione di /ST0 fa saltare immediatamente nello stato st00 e le uscite assumono il valore impostato nella tavola di verità per questo stato. La sua azione è simile a quella dell'ingresso /RESET, con la differenza che l'attivazione del /RESET porta il sistema in st00 ma le uscite vengono poste in alta impedenza.
/HOLD	15	Blocca temporaneamente le transizioni di stato dovute ai soli timeout	Ha effetto solo sullo scadere dei timeout. Se un timeout scade mentre /HOLD è basso, viene rinviato a quando /HOLD ritornerà alto. Gli ingressi IN1...IN4 e il telecomando restano attivi. Se /HOLD va basso e ritorna alto prima dello scadere di un timeout, esso non ha alcun effetto.
/STOP	14	Blocca il funzionamento del Nutchip	L'attivazione dell'ingresso /STOP blocca il funzionamento del chip nello stato in cui si trova. Gli ingressi e il telecomando sono disattivati; il timer funziona regolarmente, ma se si verifica un timeout mentre /STOP è basso, esso non verrà accettato.
/RESET	1	Riporta il funzionamento del Nutchip nello stato iniziale ST00	L'attivazione dell'ingresso /RESET porta il chip nello stato iniziale st00, azzera tutti i timeout e disattiva l'interfaccia seriale di comunicazione con il PC. Per tutto il tempo in cui /RESET è basso, le uscite sono poste in alta impedenza.

Tabella 3

Componenti

<p>NUTCHIP - NUT 01-AK dual-in-line</p>  <p>pin 1, pin 10, pin 11, pin 20</p>	<p>NUTCHIP - NUT01-DEA dual-in-line</p>  <p>pin 1, pin 10, pin 11, pin 20</p>
<p>TSOP1836 - TK1836 (Vishay - Telefunken - Temic)</p>  <p>TK1836, +5V, OUT, GND</p> <p>Infrared receiver module, 36 kHz, 5 volt.</p>	<p>MC34064 (Motorola)</p>  <p>1: RESET, 2: +5V, 3: GND</p> <p>Reset generator IC, threshold = 4.5 volt or 4,75 volt.</p>
<p>TIP122 (Fairchild)</p>   <p>TIP122</p> <p>Transistor darlington NPN, max 100V 5A.</p>	<p>7805 (Texas Instruments)</p>  <p>Vout 5V stab., Vin (7..18V)</p> <p>Series voltage regulator, output 5V 1A max..</p>
<p>BC337, BC547 e molti equivalenti NPN</p>  <p>NPN general purpose bipolar junction transistor</p>	<p>BC557 e molti equivalenti PNP</p>  <p>PNP general purpose bipolar junction transistor</p>
<p>Smartcard "Funcard Purple" Galaxy</p> 	 <p>Atmel AT90S8515 + 24LC64 ISO smart card</p>

74HC00

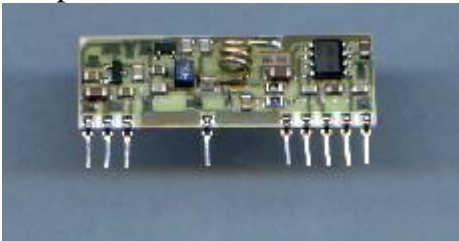


Quadruple logic NAND gate

Ricevitore radiofrequenza mod. 2-5000650 (Mipot)

Formato "in line" con dimensioni: 38,2 x 14 x 6 mm.

Pin passo 2,54 mm.



Il piedino 1 è quello più a sinistra nella foto. I piedini non usati non sono presenti, per questo la numerazione fa dei "salti"

1	+5V
2	Ground
3	Antenna
7	Ground
11	Ground
12	+5V
13	Test Point
14	Output
15	+5V

Ricevitore radiofrequenza mod. NB-CE (Aurel)

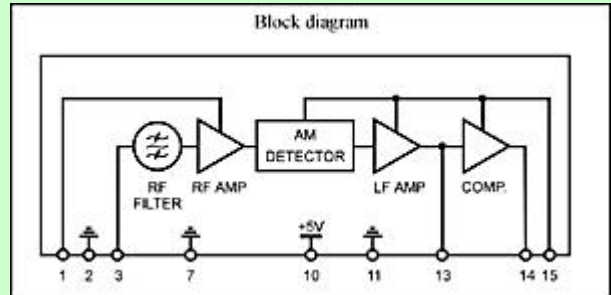
Omologato BZT

Formato "in line" con dimensioni:
44,7 x 19 x 4,7 mm.
Pin passo 2,54 mm.



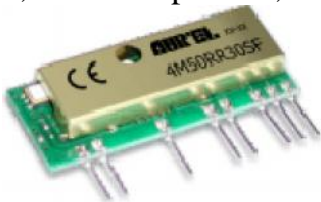
Il piedino 1 è quello più a sinistra nella foto. I piedini non usati non sono presenti, per questo la numerazione fa dei "salti"

- 1 +5V
- 2 Ground
- 3 Antenna
- 7 Ground
- 10 +5V
- 11 Ground
- 13 Test Point
- 14 Output
- 15 +5V



Ricevitore radiofrequenza mod. RX-4M50RR30SF (Aurel)

Formato "in line" con dimensioni: 40,13 x 17,51 x 5,5 mm. Pin passo 2,54 mm.



Si parte dal pin 2, quello più a sinistra nella foto. I piedini non usati non sono presenti, per questo la numerazione fa dei "salti"

2	Ground		
3	Antenna		
7	Ground		
10	+5V		
11	Ground		
13	Test Point	14	Output
		15	+5V

Ricevitore radiofrequenza mod. RX-4M50SA60SF (Aurel)

Formato "in line" con dimensioni: 44,95 x 17,51 x 9,3 mm.
Pin passo 2,54 mm.



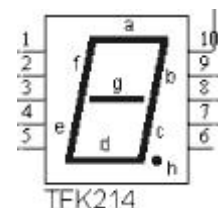
Si parte dal pin 1, quello più a sinistra nella foto. I piedini non usati non sono presenti, per questo la numerazione fa dei "salti"

1	+5V	
2	Ground	
3	Antenna	
7	Ground	
10	+5V	
11	Ground	
13	Test Point	
14	Output	
15	+5V	

Display LED 7 segmenti TFK216

pieidino	segmento
1	G
2	F
3	catodo comune

pieidino	segmento
6	H (DP)
7	C
8	catodo comune



4	E	9	B
5	D	10	A

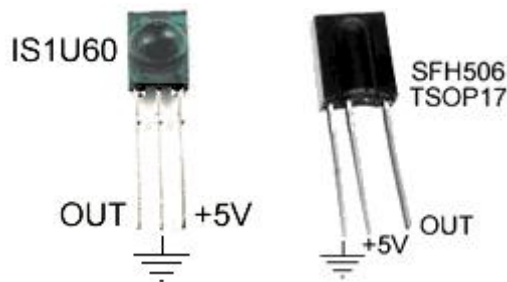
Microswitch



I microswitch sono piccoli deviatori azionati da una linguetta. I tre contatti sono: contatto comune, normalmente chiuso (NC), normalmente aperto (NO).

Si trovano nei negozi di materiale elettrico per l'industria e, molto più a buon mercato, come parti di recupero nei mercatini.

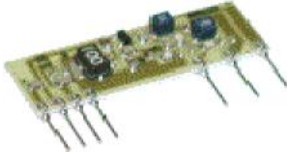
Alternative per ricevitori a raggi infrarossi



Il modulo TSOP1836 è quello che dà i migliori risultati, ma se non lo trovate potete ripiegare sullo IS1U60 (Sharp), SFH506 (Sharp), TSOP1736 (Telefunken). La frequenza del modulo deve essere di 36kHz. Un modulo da 38 o 40 kHz funzionerà ugualmente, ma con una portata *molto* ridotta!

Trasmettitore radiofrequenza mod. TX-SAW433SZ (Aurel)

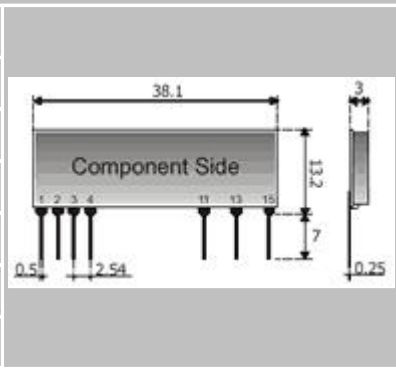
Per realizzare telecomandi.
 Formato "in line" con dimensioni:
 38,1x 13,2x 3mm. Pin passo
 2,54mm.



Si parte dal pin 1, quello più a sinistra nella foto

I piedini non usati non sono presenti, per questo la numerazione fa dei "salti"

1	Ground
2	non usare
3	Input
4	Ground
11	Antenna
13	Ground
15	+5V

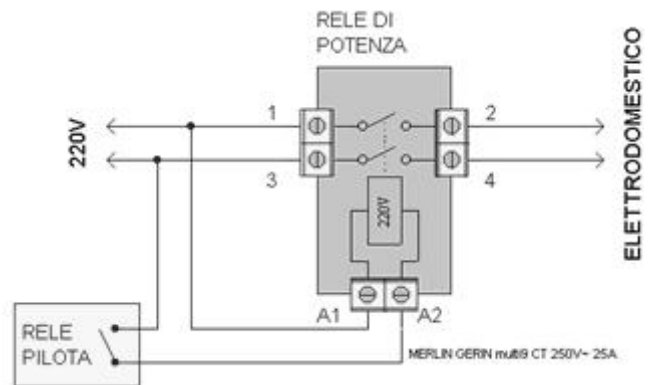


Frequenza:433,92 MHz (consigliata antenna esterna da 16,5 cm) a norme europee EN300220, EN60950

Relè di potenza per elettrodomestici



Un relè della Merlin Gerin, montabile su barra DIN. Se la bobina del relè non fosse a 220V, dovrete interporre un trasformatore.



Collegamento di un relè di potenza con bobina a 220 volt. La numerazione dei morsetti si riferisce ad un modello della Merlin Gerin.

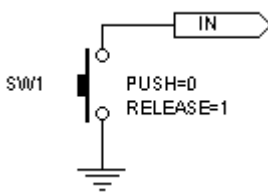
Il relè di potenza o teleruttore si utilizza per accendere un elettrodomestico ad elevato consumo (più di 2 ampere, pari a circa 400 watt), come ad esempio un forno o una lavatrice. Il relè di potenza **non** si pilota direttamente con il Nutchip, si **interpone** invece un relè più piccolo (il relè pilota). E' il relè pilota che accende il relè di potenza, secondo lo schema qui a lato.

Circuiti base

Ingressi

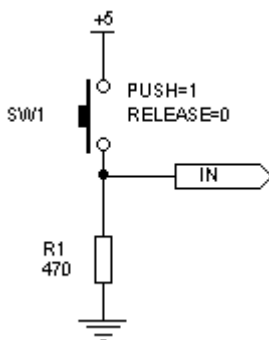
I circuiti di ingresso si applicano, oltre che agli ingressi **generici** IN1, IN2, IN3 e IN4, **anche** agli ingressi **speciali** ST0, STOP e HOLD. Se si utilizza l'ingresso IN4 bisognerà lasciare bianca la casella "**abilita comparatore analogico**".

Pulsante o interruttore (verso massa)



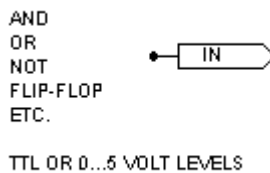
Il modo più semplice di collegare un pulsante o un interruttore al Nutchip è quello fra ingresso e massa (GND). Gli ingressi del Nutchip sono provvisti di una resistenza interna che li mantiene a livello positivo. Perciò a pulsante rilasciato l'ingresso vale 1. Premendo il pulsante si collega l'ingresso a massa, perciò si leggerà uno 0.

Pulsante o interruttore (verso il positivo +5V)

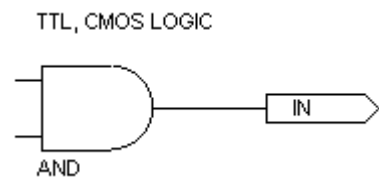


E' possibile collegare un pulsante anche verso il positivo; in tal caso è necessario contrastare l'azione della resistenza interna di *pull-up* che è incorporata nel NUTCHIP. La resistenza R1 da 470 ohm (collegata verso massa, cioè come *pull-down*) svolge proprio questa azione.

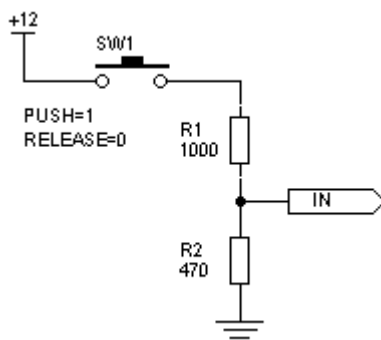
Porte logiche



Tutti gli ingressi possono essere collegati alle uscite di porte logiche TTL o porte logiche CMOS se alimentate a 5V. L'esempio a destra mostra come collegare l'uscita di una porta AND ad un ingresso del NUTCHIP.



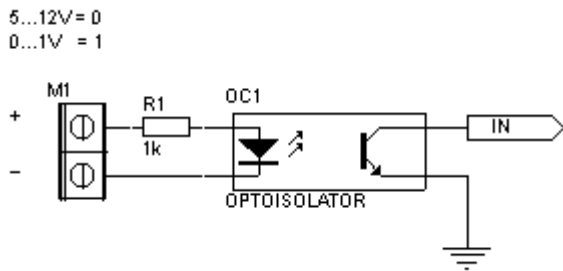
Ingressi a 12 volt (24 volt)



Se gli ingressi devono essere collegati ad un circuito che fornisce in uscita una tensione (**continua**) più alta di 5 volt, la possiamo ridurre con un partitore come in figura.

Questo circuito offre una buona **resistenza ai disturbi**, che vengono attenuati dal partitore di ingresso. E' possibile migliorare ulteriormente questa caratteristica aggiungendo un condensatore da **100 nF** in parallelo ad R2. Per tensioni di **24 volt** utilizzare per R1 il valore di **2200 ohm**.

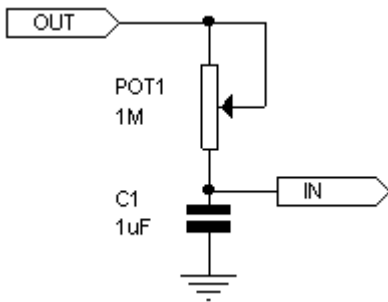
Ingresso optoisolato



Quando è necessario garantire l'isolamento galvanico fra i segnali esterni ed il circuito del NUTCHIP, si ricorre ad un optoisolatore. Con il morsetto scollegato o con una tensione minore di 1 volt il NUTCHIP legge 1. Quando il LED interno all'optoisolatore si illumina, il transistor conduce l'ingresso del NUTCHIP è condotto a GND: l'ingresso quindi vale zero.

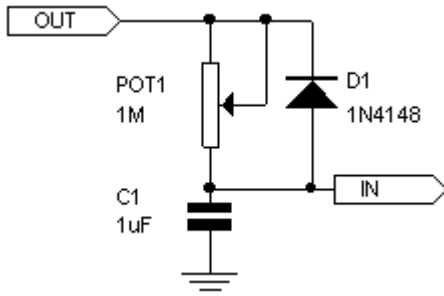
Potenzimetro (1)

- Usate questo circuito per:
-) misurare una resistenza
 -) impostare tempi regolabili



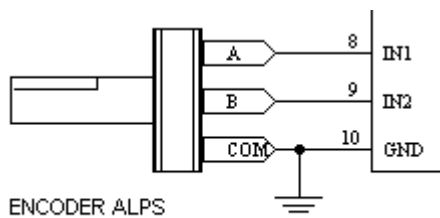
Funziona così: si mette a zero l'uscita per un tempo lungo a sufficienza per scaricare del tutto il condensatore C1. Sarà necessario attendere almeno il doppio del massimo tempo impostabile. Poi si rimette l'uscita a uno. L'ingresso tornerà a uno solo quando il potenziometro, caricando il condensatore, gli avrà fatto raggiungere almeno 2 volt. **Nota:** se il circuito viene utilizzato con il cursore a zero (resistenza nulla) è consigliabile aggiungere una resistenza da 220 ohm in serie con il potenziometro per limitare la corrente attraverso il condensatore.

Potenzimetro (2)

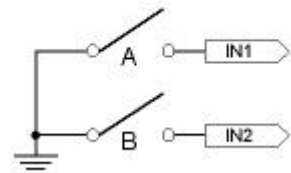


E' una versione migliorata del circuito precedente, in cui il condensatore viene scaricato quasi istantaneamente grazie al diodo D1. Il periodo di scarica, in cui si tiene bassa l'uscita, potrà essere ridotto al minimo. Per il resto funziona come il circuito precedente.

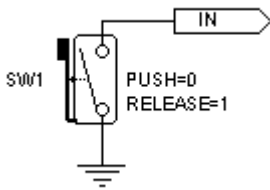
Encoder



Un encoder meccanico equivale ad una coppia di interruttori: i contatti di solito sono solo tre perchè al piedino COM è collegato un polo di entrambi gli interruttori. Alcuni modelli di encoder prevedono che uno o entrambi gli interruttori siano chiusi anche in condizioni di riposo.

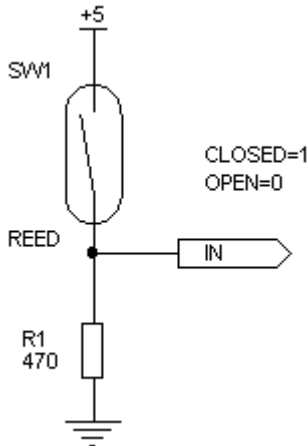


Finecorsa



Un finecorsa ("microswitch") non è altro che un tipo speciale di interruttore a deviatore. Lo si collega al NUTCHIP come se fosse un pulsante, preferibilmente verso massa o verso il positivo. Se il filo di collegamento dovesse essere lungo, meglio usare il collegamento verso il positivo a 12 volt ed inserire un condensatore per eliminare eventuali disturbi captati dal filo.

Sensore magnetico (verso il positivo)

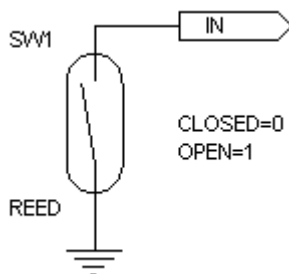


I sensori magnetici (usati spesso negli antifurto) sono dei piccoli interruttori che si aprono o si chiudono quando vengono avvicinati ad una calamita. I più diffusi sono quelli che restano chiusi a riposo (cioè quando la calamita è vicina). Con un contatto normalmente chiuso l'allarme scatta anche se si tenta di tagliare i fili, perchè il taglio apre il circuito interrompendo la corrente.

Un sensore magnetico si collega come un comune interruttore. Volendolo collegare al polo positivo, è necessario contrastare l'azione della resistenza interna di *pull-up* che è incorporata nel NUTCHIP. La resistenza R1 da 470 ohm (collegata verso massa, cioè come *pull-down*) svolge proprio questa azione.

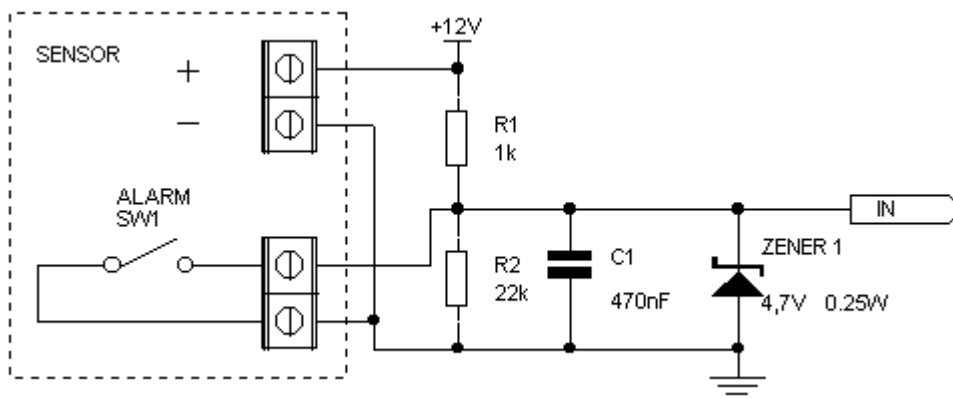
Se il filo di collegamento dovesse essere lungo, meglio usare il collegamento verso il positivo a 12 volt o quello per sensori antifurto ed inserire un condensatore per eliminare eventuali disturbi captati dal filo.

Sensore magnetico (verso GND)



Il modo più semplice di collegare un sensore magnetico al Nutchip è quello fra ingresso e massa (GND). Gli ingressi del Nutchip sono provvisti di una resistenza interna che li mantiene a livello positivo. Perciò a contatto aperto l'ingresso vale 1. Chiudendo il contatto si collega l'ingresso a massa, perciò si leggerà uno 0. Se utilizzate il sensore magnetico **in un antifurto**, allora è meglio usare il modello normalmente chiuso con il circuito che segue, una versione che migliora parecchio la tenuta ai disturbi captati dai fili.

Sensore antifurto



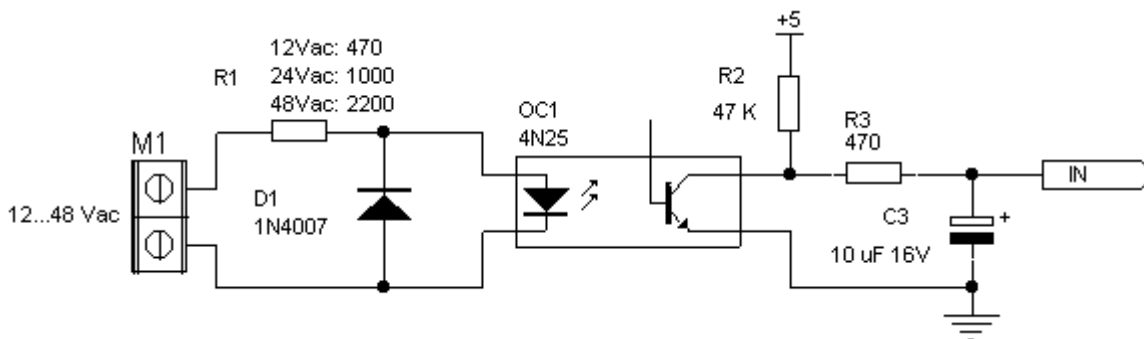
I sensori antifurto hanno questi morsetti:

-) un'alimentazione a 12 volt (proveniente dalla centrale)
-) un contatto normalmente chiuso, che si apre quando il sensore scatta
-) talvolta un secondo contatto

di sicurezza (tamper 24 ore) che si apre quando si apre il coperchio

Con un alimentatore a doppia uscita (+12V per il sensore, +5V per il NUTCHIP, stesso negativo per entrambi) questo schema permette di collegare e alimentare il sensore con solo tre fili. Inoltre la presenza di un generoso condensatore e del diodo zener spianano eventuali disturbi captati da fili di collegamento, che di solito sono piuttosto lunghi.

Ingresso in corrente alternata



A volte occorre prelevare segnali in corrente alternata:

) per "sentire" un campanello esistente: i trasformatori per campanelli sono di solito a 12 o a 24 volt
) per collegarsi al quadro elettrico di controllo di una macchina per

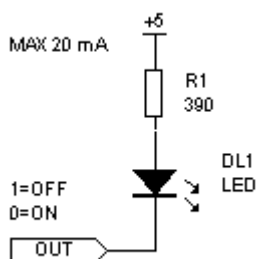
"sentire"
fine corsa,
galleggianti,
,
teloruttori,
microswitch
h
esistenti

Questo schema è adatto per tensioni alternate da 12 fino a un massimo di 48 volt, cambiando il valore della resistenza R1.

Uscite

I circuiti che seguono si applicano alle uscite del NUTCHIP: OUT1, OUT2, OUT3 e OUT4. Tipici dispositivi di uscita sono LED, relè, cicalini, ma non mancano neppure porte logiche o display a 7 segmenti. Per il collegamento di lampade o motori un relè è la soluzione più semplice e diretta

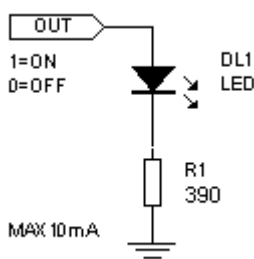
LED (1)



Questa è la configurazione che di solito si preferisce per pilotare dei LED con delle uscite digitali. Si tratta di un'eredità "storica", quando le uscite digitali erogavano più corrente quando erano a livello logico zero di quanto non riuscissero fare quando erano a livello uno. Il LED si accende quando l'uscita è a zero (assorbimento di corrente) e si spegne quando l'uscita è a uno (corrente pressochè nulla).

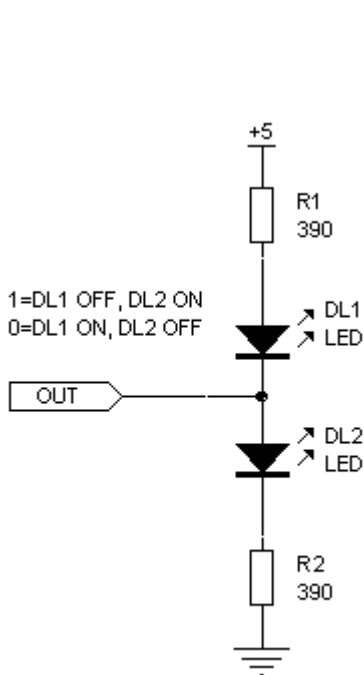
Collegando il LED con il catodo (la parte fatta a freccia) verso il polo positivo la corrente scorre quando l'uscita si trova a zero. La resistenza limita la corrente che scorre attraverso il LED e che transita sull'uscita. Con una resistenza da 390 ohm si ottiene una corrente di circa 10mA, un valore ottimale per il pilotaggio di un LED.

LED (2)



Se si preferisce che il LED si accenda quando l'uscita è a uno anzichè quando è a zero si collega il catodo del LED all'uscita e l'anodo a GND attraverso una resistenza. La resistenza sarà da 390 ohm perchè in configurazione positiva il NUTCHIP può erogare al massimo 10 mA.

LED Doppio e LED bicolore



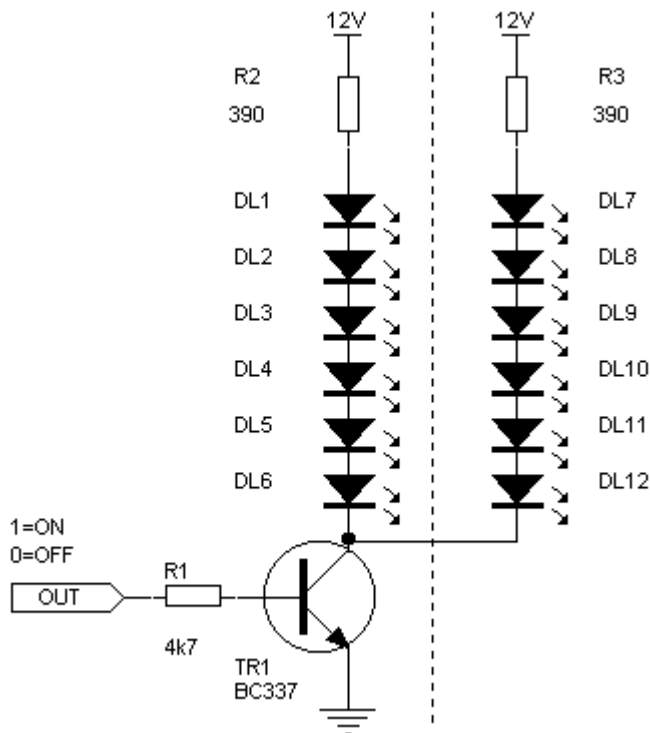
Con questo collegamento si hanno due LED collegati ad una sola uscita, in modalità tale che uno esclude l'altro. E' l'ideale per pilotare LED di colori diversi, ad esempio mettendo un LED verde per DL1 ed uno rosso per DL2 abbiamo che la luce cambia colore a seconda dello stato dell'uscita. Alcuni LED bicolore presenti in commercio contengono al loro interno una coppia di diodi LED collegati esattamente come DL1 e DL2, cioè con una coppia catodo-anodo in comune, e si possono usare per questo circuito.

Quando l'uscita è alta, il LED DL2 è acceso ed il LED DL1 è spento. Se invece l'uscita del Nutchip diventa bassa, sarà il LED DL1 ad accendersi, mentre il LED DL2 si spegnerà.

Le resistenze sono entrambe da 390 ohm. La corrente totale sarà sempre di 10 mA circa (e non di 20 mA come si potrebbe a prima vista supporre sommando la corrente di R1 con quella di R2) perchè è acceso sempre e solo un LED per volta.

Nota: Un comportamento interessante di questo circuito è che se provate a collegare solo i LED e le resistenze, ma senza collegare il circuito al piedino di uscita del Nutchip, allora vedrete accendersi entrambi i LED ad intensità ridotta!

Gruppi di LED

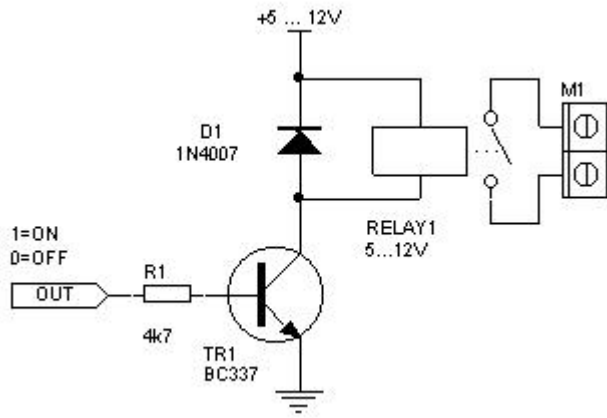


Per realizzare giochi di luce, decorazioni natalizie, scritte lampeggianti, è necessario accendere molti LED per volta.

Questo circuito può pilotare fino a 12 LED per ogni uscita (potete però montare anche solo una delle due catene di 6 led separate dal tratteggio). Se una della catene di LED dovesse avere meno di 6 LED, la resistenza in serie dovrà essere aumentata rispetto agli attuali 390 ohm. in pratico la corrente che la attraversa dovrà essere di circa 10 mA per non sovraccaricare i LED.

N. di LED	Resistenza
5	470
4	680
3	750
2	820
1	1000

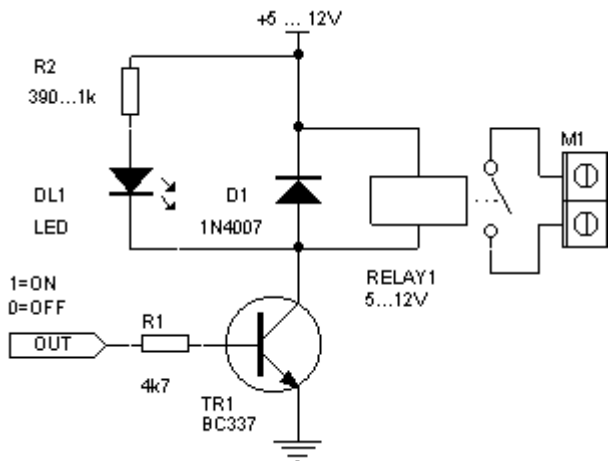
Relè



Assieme ai LED, i relè sono i dispositivi più usati assieme ai Nutchip. Tramite il relè il Nutchip può comandare luci, motori elettrici, elettrovalvole, trasformatori, riscaldatori, frigoriferi, pompe, e mille altri dispositivi. Se il relè è da 5 volt, lo collegheremo alla stessa alimentazione del NUTCHIP ; se invece ha una tensione maggiore (9 o 12 volt) ci servirà un alimentatore con doppia tensione. Non è necessario che l'uscita a 9 o 12 volt sia stabilizzata.

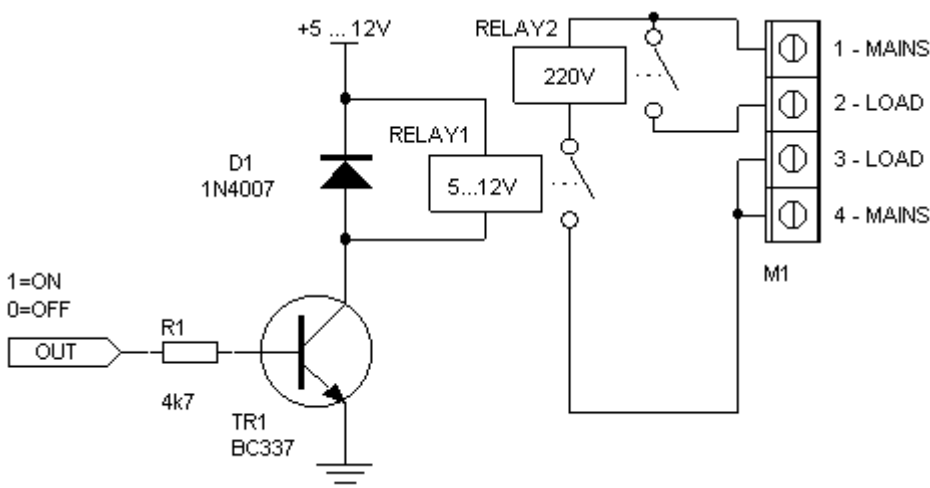
Il relè ha il grande vantaggio di isolare il carico esterno (collegato al morsetto M1) dal circuito elettronico di comando (Nutchip). **Attenzione:** siccome spesso il carico è alimentato con tensioni pericolose per l'uomo, è necessario porre la *massima attenzione* per evitare che i contatti del relè e del morsetto M1 entrino in contatto con altre parti del circuito.

Relè con indicazione a LED



E' **molto comodo** avere un LED in modo che si accende quando il relè è eccitato. Utilizzando un relè da 5 volt montare una resistenza R2 da **390 ohm**. Per 9 volt utilizzare una **820 ohm** e per 12 volt una **1000 ohm**.

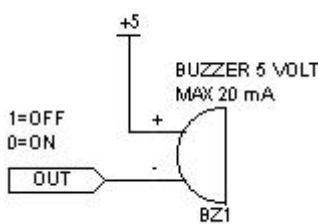
Relè di potenza a 220V



I relè di potenza a 220 volt (Relay 2 nello schema) ed i teleruttori si collegano tramite il contatto di un relè pilota di minor potenza (Relay 1). Se l'ampereaggio di Relay 1 può non essere elevato, la tensione alternata applicabile ai suoi contatti deve essere maggiore di

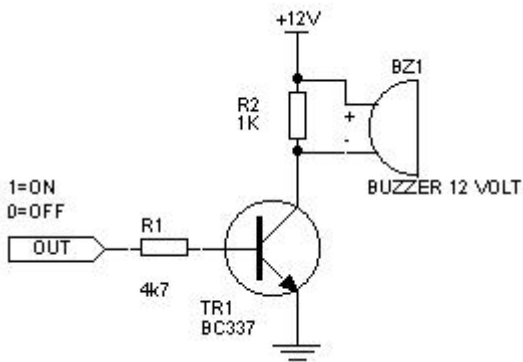
250V. La rete (mains) si applica ai morsetti 1 e 4; il carico (ad esempio un motore elettrico o una lampada) invece ai morsetti 2 e 3. **Pericolo:** fate la massima attenzione nell'effettuare questi cablaggi per via della presenza della tensione di rete a 220 volt.

Buzzer a 5 volt



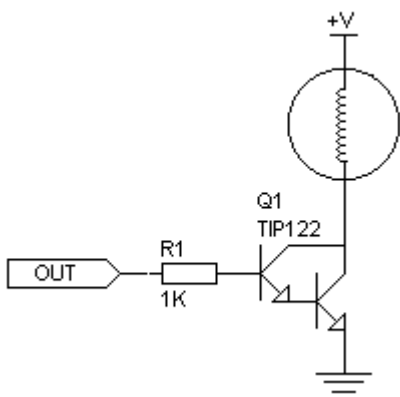
Un cicalino a 5 volt può essere pilotato direttamente da una delle uscite del NUTCHIP. Bisogna sempre scegliere i cicalini di tipo elettronico, in quanto i cicalini elettromeccanici assorbono molto più dei 20 mA massimi disponibili. La maggior parte dei cicalini a 9 volt funziona ugualmente bene a 5 volt, sebbene ad intensità ridotta.

Buzzer a 12 volt e buzzer di potenza



I buzzer a 12 volt sono i più diffusi. A volte il livello sonoro è tale da potere parlare di vere e proprie minisirene. Il circuito illustrato è adatto per buzzer fino a 100 mA di assorbimento. Per dispositivi più potenti prendete in considerazione l'utilizzo di un relè.

Lampade a bassa tensione

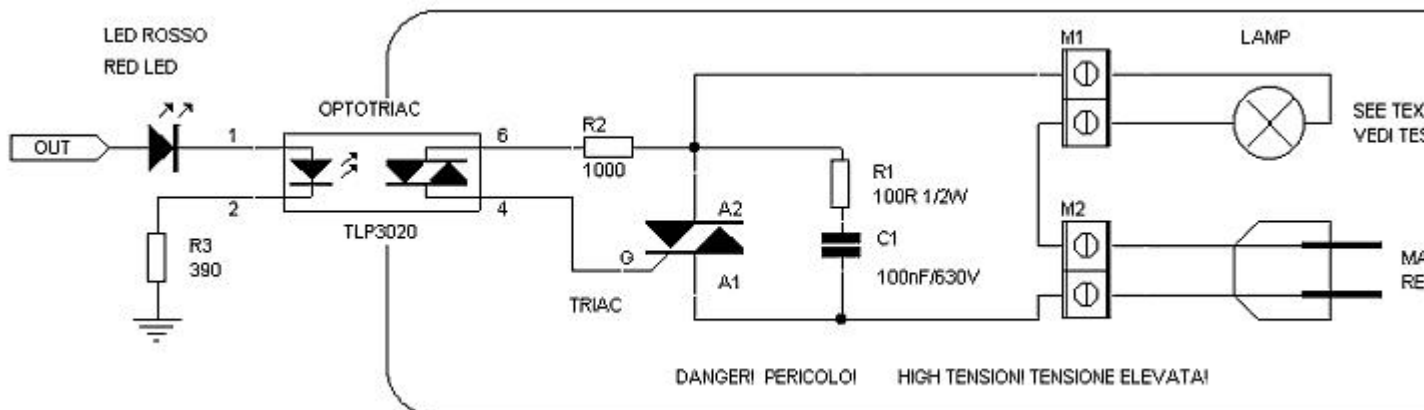


Le lampade da 3 fino a 24 volt funzionanti in corrente continua si pilotano tramite un transistor darlington di potenza (il simbolo è quello di due transistor!), come il TIP122. La tensione +V è quella specificata per le lampade, e non deve superare i 40 volt. Suggerimento: se si pilota l'uscita con dei veloci impulsi on/off si può modulare l'intensità luminosa della lampada.

Lampade a 220V con optoTRIAC

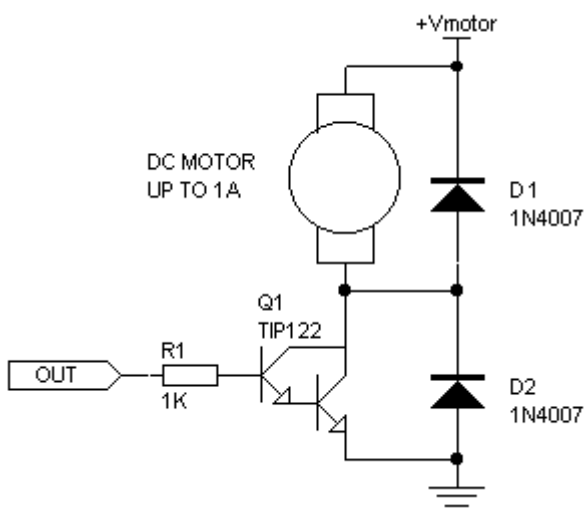
Nei giochi di luce si utilizzano quasi sempre i TRIAC. Un relè infatti non è indicato per azionarsi più volte in un secondo, e a lungo andare si romperebbe. Questo schema utilizza un optoTRIAC per isolare il circuito del NUTCHIP dalla parte di

circuito soggetto alla tensione di rete (tutta la zona con l'indicazione di "PERICOLO!"). Un secondo TRIAC, eccitato dall'optoTRIAC, pilota la lampada. Questo circuito si può usare anche per lampade a corrente alternata a 12 e 24 V (come le lampade alogene), ma NON per pilotare motori, trasformatori, lampade al neon.



Normalmente si usano TRIAC da almeno 600V e 5A, con i quali si potrà pilotare un carico massimo di 200W a 220Vca. Per carichi più grandi sarà necessario utilizzare un TRIAC a maggior corrente e montarlo su una aletta di raffreddamento **accuratamente isolata**.

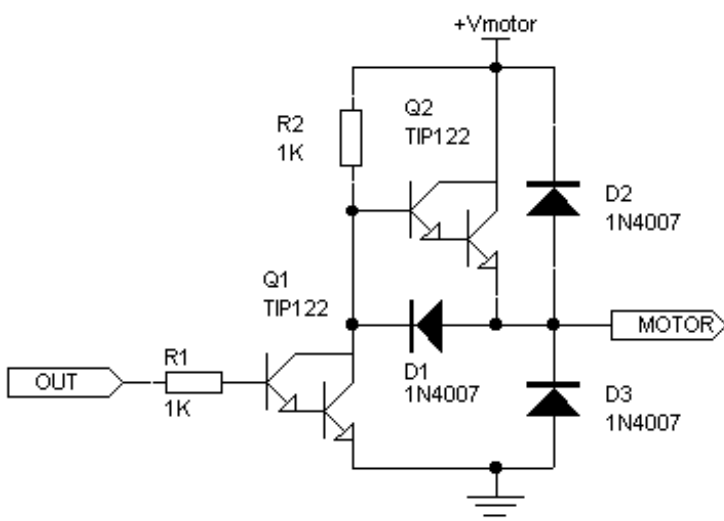
Motori ON/OFF



Basta un transistor di tipo Darlington per pilotare un motore in corrente continua fino a 48 Volt. Se l'assorbimento è minore di 600 mA, non servirà l'aletta di raffreddamento. Il motore si aziona se l'uscita = 1, si spegne se l'uscita = 0.

Nota: il simbolo di un transistor Darlington assomiglia a due transistor in cascata, ma si tratta di un solo componente.

Motori: schema base (mezzo ponte)



Questo schema, più elaborato, è la base per pilotaggi più sofisticati:

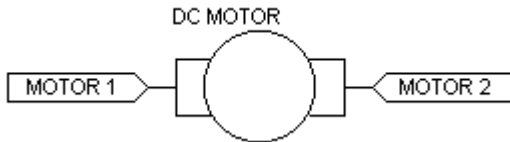
-) motori in continua con frenatura
-) motori in continua con inversione
-) motori passo-passo

Per ogni azionamento serviranno uno più di questi circuiti, ciascuno pilotato da una differente uscita del NUTCHIP.

Questo circuito viene usualmente chiamato *mezzo ponte H*.

Motori avanti/stop/indietro

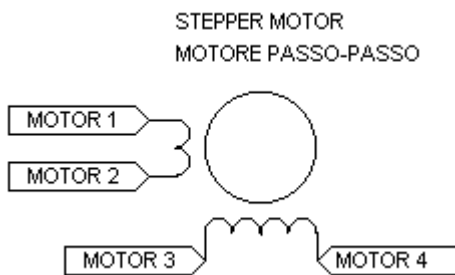
Per invertire il senso di marcia di un motore DC serve un ponte H, che si realizza con due circuiti a [mezzo ponte](#) illustrati sopra. Ogni circuito fornisce uno dei due fili MOTOR. Due uscite del Nutchip sono necessarie per pilotare le due metà del ponte. Le combinazioni delle uscite danno luogo ai seguenti azionamenti:



<i>motor 1</i>	<i>motor 2</i>	<i>movimento</i>
0	0	stop
0	1	avanti
1	0	indietro
1	1	stop

Motori passo-passo

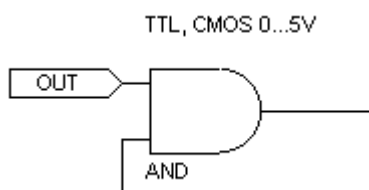
Questo circuito è adatto per stepper bipolari a 4 fili, che richiede 4 circuiti [mezzo ponte](#). Ecco la sequenza di pilotaggio:



<i>motor1</i>	<i>motor 2</i>	<i>motor 3</i>	<i>motor 4</i>
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1

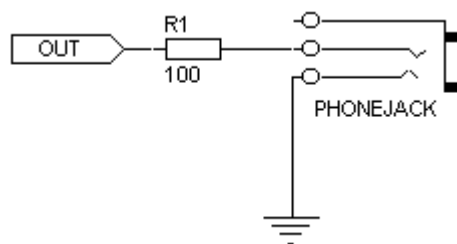
Per invertire la rotazione, basta eseguire la sequenza al contrario

Porte logiche



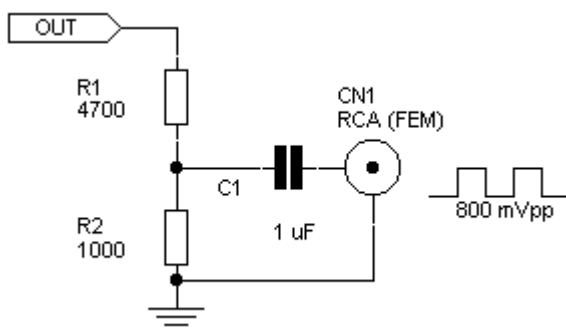
Tutti le uscite possono pilotare gli ingressi di porte logiche TTL o porte logiche CMOS se alimentate a 5V. Questo esempio mostra come collegare il NUTCHIP all'ingresso di una porta AND. Se il circuito logico è esterno, ricordate di collegare in comune i GND.

Cuffie stereo



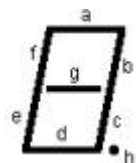
Questo circuito pilota le comuni cuffiette per walkman. Notate che il collegamento più esterno è lasciato libero. In questo modo i due padiglioni sono in serie fra di loro e suoneranno entrambi. Onde prodotte coi ritardi più brevi (1mS) suoneranno come toni, mentre aumentando i ritardi nell'ordine dei secondi udiremo degli schiocchi: se dovessero essere troppo forti aumentate il valore della resistenza.

Amplificatore audio



Questo circuito inietta il segnale nell'ingresso ausiliario degli impianti stereo. Se il vostro amplificatore è sprovvisto di questa entrate, potete utilizzare quella per un CD, una piastra (TAPE) o un sintonizzatore (TUNER) esterni. Alla presa RCA collegate l'apposito cavetto reperibile nei negozi di Hi-Fi. Lo schema si riferisce ad un solo canale (LEFT o RIGHT), ma potete collegarli entrambi mettendo una seconda presa RCA in parallelo alla prima.

Display LED



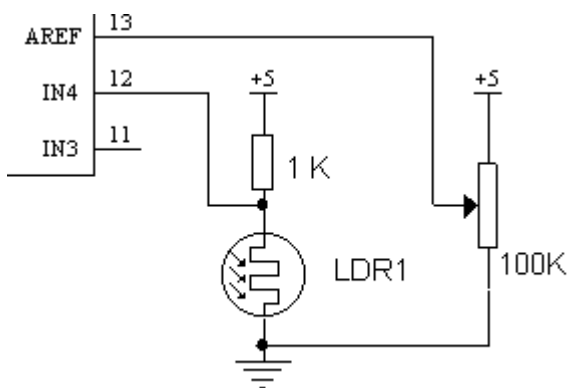
Un display costituito da 8 LED, ognuno dei quali accende un diverso "segmento" della cifra. I segmenti si indicano con le lettere a,b,c,d,e,f,g, mentre h corrisponde al puntino. Per diminuire il numero dei piedini, tutti i catodi (o gli anodi) sono connessi allo stesso pin: parliamo perciò di display a "catodo comune" (oppure "anodo comune").

Sebbene i segmenti si possano illuminare singolarmente allo stesso modo di un LED (basta mettere una resistenza in serie), di solito si utilizza un integrato apposito di decodifica, come ad esempio il CD4511 utilizzato in questo [progetto](#).

Ingresso analogico

L'ingresso IN4 del NUTCHIP può funzionare da comparatore analogico. Il confronto avviene fra la tensione presente al piedino IN4 e quella sul piedino AREF. Per utilizzarlo in questa veste occorre attivare la casella "Abilita il comparatore" al momento della programmazione.

Fotocellula (LDR = light dependant resistor)

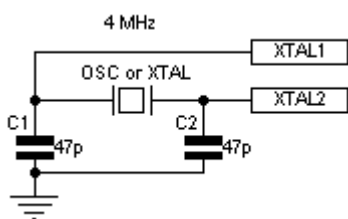


Con questo circuito l'ingresso IN4 vale 0 oppure 1 a seconda che la fotocellula sia illuminata o al buio. Se la fotocellula deve funzionare in presenza di lampade a scarica o luci al neon, si può aggiungere un condensatore da 100 nF in parallelo al LDR per ridurre le fluttuazioni generate da questo tipo di lampade. Il trimmer regola il punto di intervento della fotocellula, e va regolato in base alla luminosità dell'ambiente. Il valore del trimmer non è critico, potendo variare da 1k fino a 470k..

Oscillatori

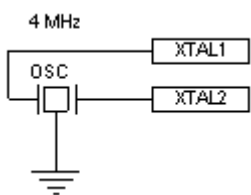
Il NUTCHIP necessita di un clock 4MHz per funzionare. L'oscillatore contenuto nel chip richiede un quarzo od un risuonatore ceramico esterni per funzionare.

Oscillatore al quarzo / Oscillatore con risonatore ceramico a 2 piedini



Quarzi ed oscillatori ceramici a 2 piedini si collegano allo stesso modo. I quarzi sono indispensabili per costruire timer per lunghi periodi sufficientemente precisi. I condensatori impediscono che l'oscillatore si inneschi ad una frequenza più alta dei 4 MHz richiesti. Può capitare di imbattersi in un quarzo che richieda condensatori di 100pF. I quarzi hanno un tempo di partenza più lungo degli oscillatori ceramici, perciò quando si utilizza il quarzo occorre sempre prevedere un circuito di reset RC o con l'apposito integrato.

Oscillatore con risonatore ceramico a 3 piedini



Il risonatore da 4 MHz a tre piedini non richiede componenti supplementari per oscillare. Il piedino da connettere a massa è quello centrale. La precisione della frequenza è solitamente migliore dell'1%. Per i brevi periodi questo errore è trascurabile, ma per periodi lunghi e ripetitivi diventa sensibile. Un orologio che avanza di un minuto ogni cento accumula un errore di più di tre giorni in un anno!

Modulo oscillatore /oscillatore esterno

Se disponete di moduli oscillatori al quarzo (si trovano talvolta su schede di recupero) potete utilizzarli collegandone l'uscita al piedino XTAL1 del NUTCHIP. Il pin XTAL2 resta scollegato.

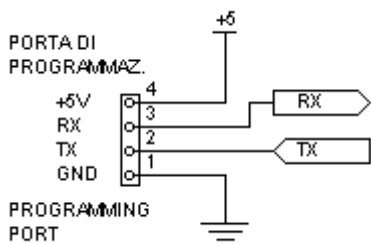
Allo stesso modo potete utilizzare il segnale da una sorgente di clock esterna.

Interfaccia per PC

Il NUTCHIP dialoga col PC tramite i piedini RX e TX. Utilizza un protocollo seriale a 9600 bps, N, 8, 1. Se vogliamo riprogrammare il NUTCHIP una volta montato sul circuito, dovremo prevedere il connettore per l'interfaccia PC oppure un'interfaccia PC completa (si tratta di 2 soli transistor) sulla scheda stessa.

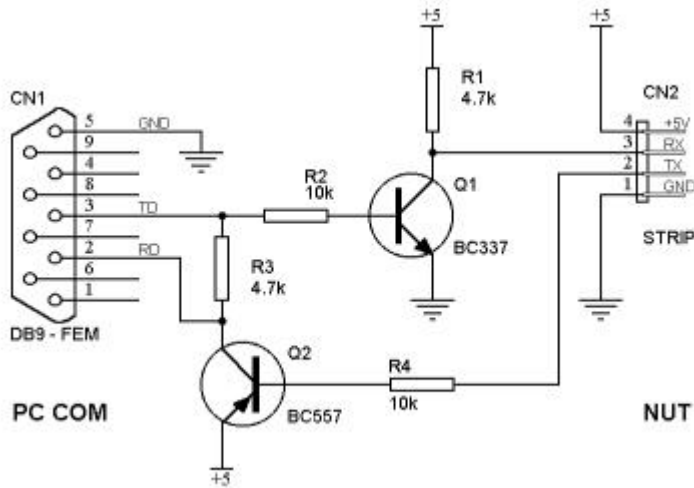
Per ulteriori informazioni sulle interfacce: [cliccare qui](#)

Connettore di programmazione



Su questo connettore si innesta l'interfaccia a transistor per il PC. Il connettore accoglie una strip maschio passo 2,54 mm che è montata sull'interfaccia. E' quindi possibile montare un connettore box femmina a 4 poli, una strip femmina o addirittura metà di uno zoccolo per integrati da 4+4 pin.

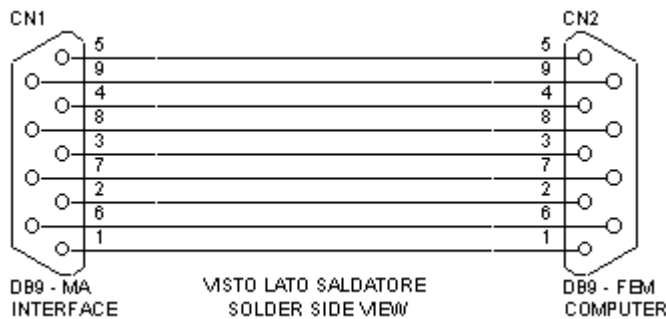
Interfaccia a transistor



E' l'interfaccia più semplice ed economica. Si può utilizzare praticamente qualsiasi transistor PNP ed NPN per piccoli segnali. Il connettore CN2 è una strip passo 2,54 che si innesta nel *connettore di programmazione*. Se il circuito viene utilizzato **senza** che il PC sia connesso (come accade se si incorpora l'interfaccia nel circuito finale), è consigliabile porre una resistenza da 10k fra la base del transistor Q1 e la massa.

[Maggiori dettagli](#)

Cavo di collegamento al PC

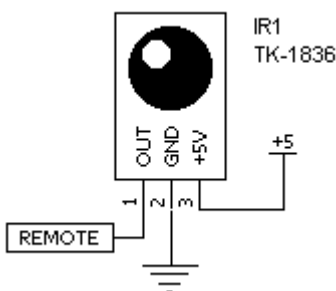


Un cavo di prolunga seriale maschio-femmina collega l'interfaccia al computer (sia che essa sia indipendente sia che venga incorporata sulla scheda). Se utilizzate l'interfaccia a transistor consigliata potrete collegare soltanto i fili 2, 3 e 5.

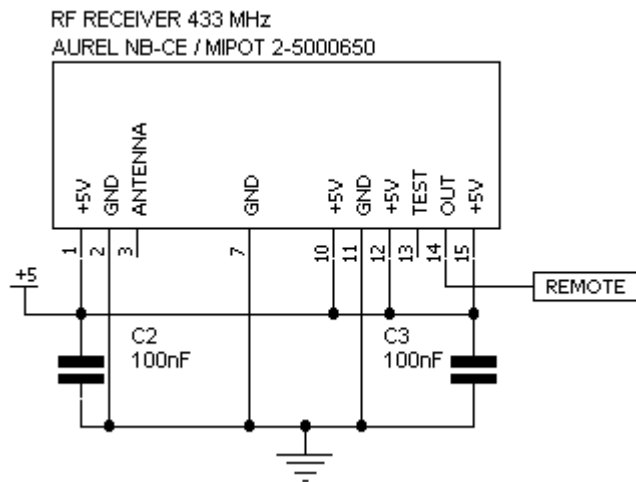
Ricevitori per telecomando

*Il segnale del telecomando va applicato al piedino REMOTE. Se il piedino non si utilizza, potremo **ancorarlo** al positivo o a massa, per impedirgli di captare disturbi. Ricordate sempre di programmare il tipo di telecomando adeguato (RF o infrarossi) al momento della programmazione del chip.*

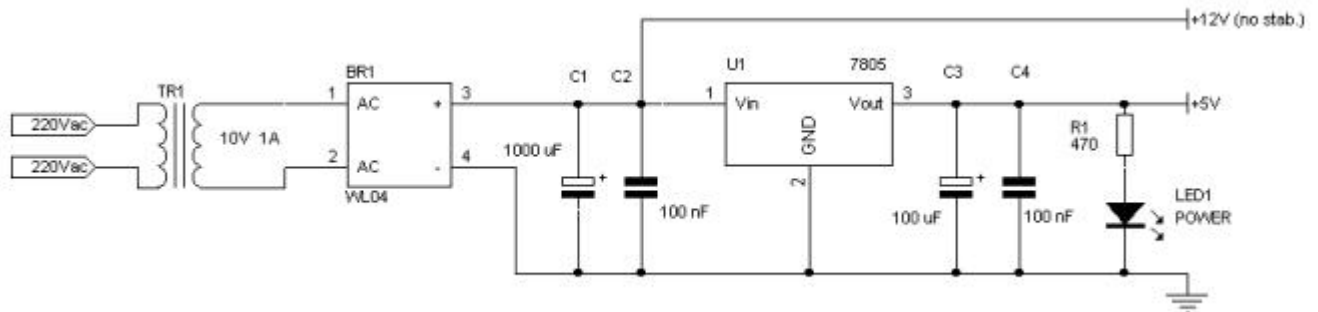
Ricevitore telecomando a infrarossi



La figura si riferisce al ricevitore TSOP1836, marcato TK1836, di produzione Temic/Telefunken/Vishay. Altri ricevitori utilizzabili sono SFH5110 (Siemens, Sharp) LT1059 (Liteon), che però hanno una piedinatura **diversa**.

Ricevitore telecomando a radiofrequenza

I migliori risultati si ottengono con i moduli ricevitori Aurel NB-CE, RX-4M50RR30SF, RX-4M50SA60SF, ma il modulo MIPOT 2-5000650 è più economico. Se vogliamo riutilizzare il modulo potremo inserirlo in una lunga strip femmina. Il pin 5 è presente solo sul modulo Aurel, il 12 solo sul Mipot. Al pin ANTENNA potremo collegare uno spezzone di filo rigido di 20...40 cm di lunghezza.

AlimentatoriAlimentatore 5V + 12V

Questo alimentatore fornisce, oltre alla tensione di 5 volt stabilizzata per il Nutchip, anche una tensione non stabilizzata di 12 volt per alimentare ad esempio i relè collegati come dall'apposito schema (vedi).

Se non usate l'uscita a 12 volt allora potrete utilizzare qualsiasi trasformatore con uscita fra 7,5 e 14 volt invece di quello da 10V indicato sullo schema. L'integrato U1 va raffreddato con un'aletta. Prestate la massima attenzione ai collegamenti del trasformatore, che funziona a 220 volt e quindi sottoposto a tensioni potenzialmente letali. E' consigliabile proteggere il circuito con un fusibile ed un interruttore.

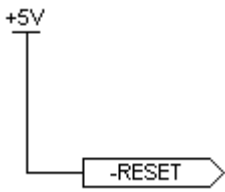
Alimentatore recuperato da PC

I mobili per PC usati o di vecchio tipo sono venduti a prezzi convenienti, e quasi sempre comprendono l'alimentatore originale. E' un alimentatore **molto potente**, capace di erogare decine di ampere, e che fornisce le tensioni di +5 volt, +12 volt (ed anche -12 volt e -5 volt). Cercate possibilmente un modello con interruttore incorporato. Normalmente i cavi che escono sono tutti in parallelo, il cavo nero a massa, il cavo rosso a +5V ed il cavo giallo a +12V (ma controllate sull'etichetta che questi collegamenti corrispondano). Se non avete problemi di spazio, utilizzate il mobile per intero.

Reset

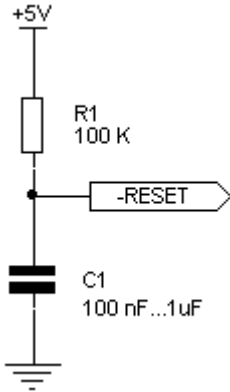
Avere un buon RESET è importante. Se un RESET improprio impedisce al chip di partire, le anomalie che possono presentarsi durante il funzionamento possono bloccare il chip se non sono accompagnate da un opportuno impulso di RESET. Un tipico esempio è il *brown-out*, cioè quando la tensione di alimentazione scende sotto il livello minimo di funzionamento pur non andando a zero. Durante il *brown-out* il RESET deve essere tenuto a zero, altrimenti il chip può perdere il controllo, bloccarsi, e addirittura richiedere una nuova programmazione.

Reset sperimentale



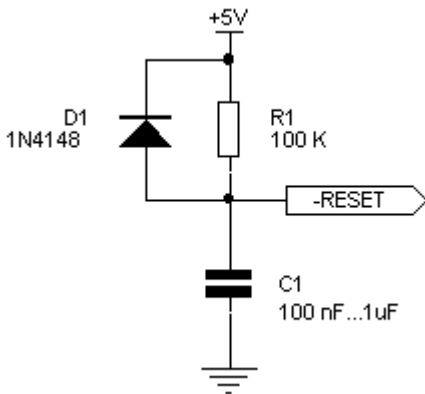
Per rapidi prototipi e per progetti che abbiano funzione di gioco e non debbano essere usati ininterrottamente per lunghi periodi, il piedino di RESET può essere semplicemente portato al positivo. Per farlo potete usare anche una resistenza da 100 kohm per sfruttare la piccola capacità parassita sempre presente sul piedino.

Reset RC



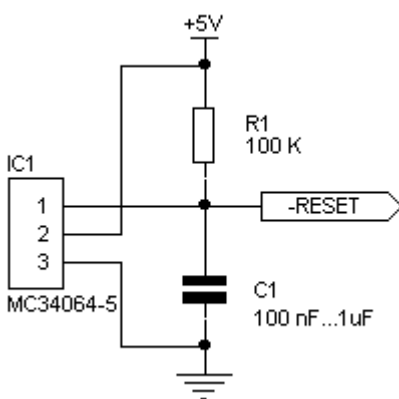
Una resistenza ed una capacità esterna costituiscono un circuito facilmente realizzabile ed assieme efficace. Il diodo (opzionale) aiuta il condensatore a scaricarsi rapidamente in caso di un "buco" di tensione. Infatti la condizione di instabilità più pericolosa è quando il RESET e l'alimentazione non sono nè alti nè bassi.

Reset RCD

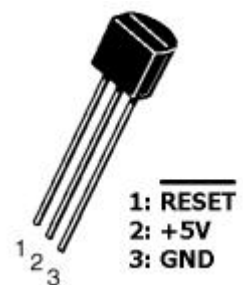


E' una versione migliorata del circuito precedente, che funziona meglio in presenza di interruzioni (o micro-interruzioni) dell'alimentazione. Il diodo D1 infatti provvede a scaricare velocemente la capacità C1 appena la tensione di alimentazione scende.

Reset a integrato



Un'integrato appositamente progettato per il RESET è ovviamente la soluzione migliore. L'MC34064 inoltre interviene con precisione scaricando il condensatore quando la tensione si abbassa, anche di poco, sotto i 4,5 volt. Quando la tensione di alimentazione non è ancora stabile e quindi sotto i 4,5 volt (succede tutte le volte che accendiamo o spegniamo l'apparecchio, ed ad ogni black-out) possiamo essere certi che il piedino di RESET viene mantenuto basso.



lampeggiatore a led

Schema elettrico

Per fare lampeggiare un LED ci basta una sola uscita. Collegheremo il LED all'uscita 4 (OUT4) del NUTCHIP, portando l'altro capo ai +5V tramite una resistenza da 390 ohm.

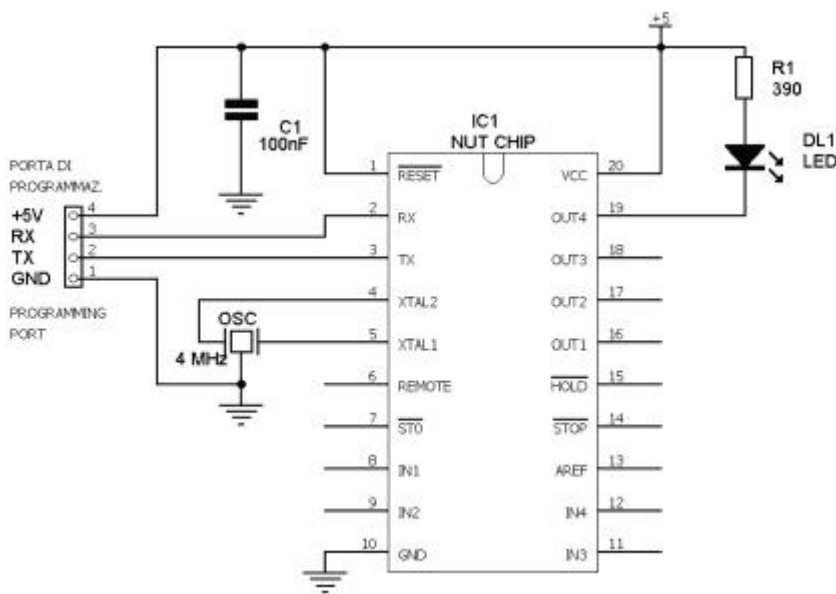
Collegando il LED al **positivo**, cioè con l'**anodo** (A) ad una resistenza che va al positivo ed il **catodo** (K) ad un piedino di uscita del NUTCHIP, la corrente vi scorrerà quando l'uscita si trova a livello zero, cioè collegata a GND, il polo negativo dell'alimentazione.

Poichè a livello **zero un'uscita può assorbire** fino a 20 mA, non ci saranno problemi ad assorbire i pochi mA richiesti dal LED per fornire luce piena. Notate che contrariamente a quanto saremmo portati a pensare, il LED si **accende** con l'uscita a livello **0** e si **spegne** con l'uscita a livello **1**.

Un condensatore da 100 nF fra positivo e negativo spianerà gli eventuali disturbi captati dalle linee di alimentazione. Il Nutchip oscilla per mezzo di un risonatore ceramico a tre piedini collegato ai piedini OSC. Questo genere di oscillatore va a regime molto in fretta, per cui non c'è bisogno di ritardare il **RESET**, che collegheremo **direttamente al positivo**.

Per il momento, lasceremo scollegati tutti gli ingressi inutilizzati del NUTCHIP.

Schema elettrico del lampeggiatore a LED



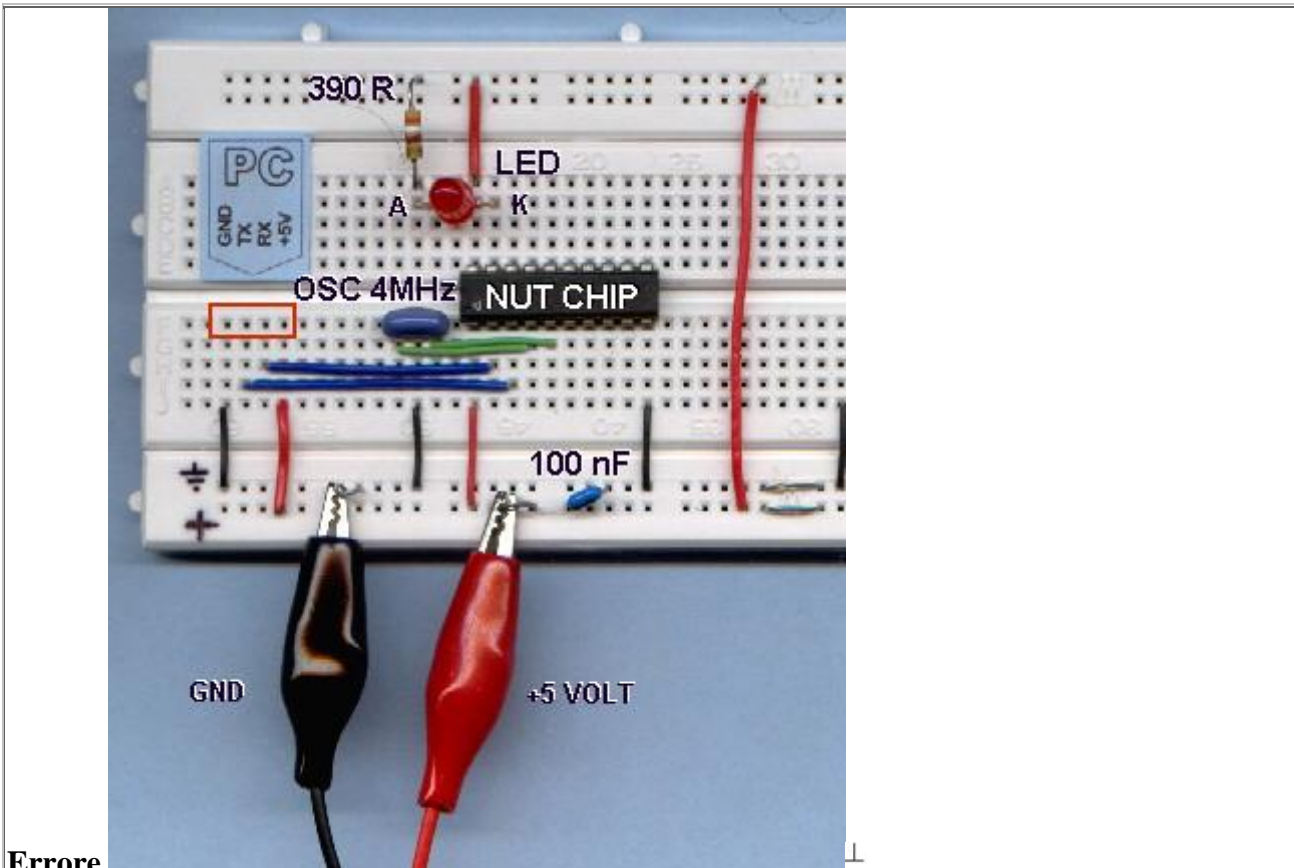
Elenco componenti:

-) 1 diodo LED
-) 1 resistenza da 390 ohm
-) 1 risonatore ceramico da 4 MHz a tre piedini
-) 1 condensatore da 100 nF
-) 1 Nutchip

Occorrono inoltre: un alimentatore a 5V, una basetta sperimentale con contatti a molla, l'interfaccia Nutchip completa di cavo, vari spezzoni di filo rigido sottile di vari colori, e naturalmente un PC con il software Nutstation.

Montaggio

Questo schema è molto semplice, e lo realizziamo facilmente su una basetta sperimentale con contatti a molla. Fate attenzione a non invertire il verso di inserzione del Nutchip, che nella foto ha il piedino 1 in basso a sinistra, vicino all'oscillatore ceramico. Il piedino 1 è identificato da una tacca o un puntino sul corpo dell'integrato; guardando il chip da sopra, gli altri piedini sono numerati progressivamente, in senso antiorario. Anche il LED ha un polarità che non si deve invertire: il catodo (K) si riconosce dal terminale più corto e dallo smusso sul contenitore.



Errore.

Nei 4 contatti incorniciati in rosso inseriremo l'interfaccia PC per programmare il NUTCHIP :

Per le connessioni usate del filo rigido sottile, possibilmente di vari colori. Usate sempre filo ROSSO per il POSITIVO (+5V) e filo NERO per il NEGATIVO (GND)

La tavola della verità

Il Nutchip che abbiamo montato non è ancora in grado di funzionare: infatti dobbiamo inserire al suo interno la **tavola degli stati** che stabilisca la relazione fra ingressi e uscite.

La tavola degli stati è un particolare tipo di *truth table*, ossia una tavola della verità.

Nella tavola della verità di un Nutchip per ognuno degli **stati** in cui si viene a trovare il nostro progetto, sono scritte una **serie di una o più condizioni** che stabiliscono se dobbiamo rimanere nel medesimo stato o se si debba passare ad un altro stato della tabella. Cambiare stato è il modo utilizzato dai Nutchip per "*fare muovere*" le uscite, perchè ad **ognuno degli stati** corrisponde una **configurazione** a nostro piacere dei livelli logici **delle uscite**.

Ma in quanti stati si può trovare un lampeggiatore? Gli stati di un lampeggiatore **sono due**: LED **spento** e LED **acceso**. Partendo dallo stato zero (in cui il LED è spento), si attende **mezzo secondo** e poi si **passa** automaticamente allo stato uno (LED acceso); da qui si

attende un altro **mezzo secondo** e poi si **passa** di nuovo allo stato zero (LED spento), e così via all'infinito.

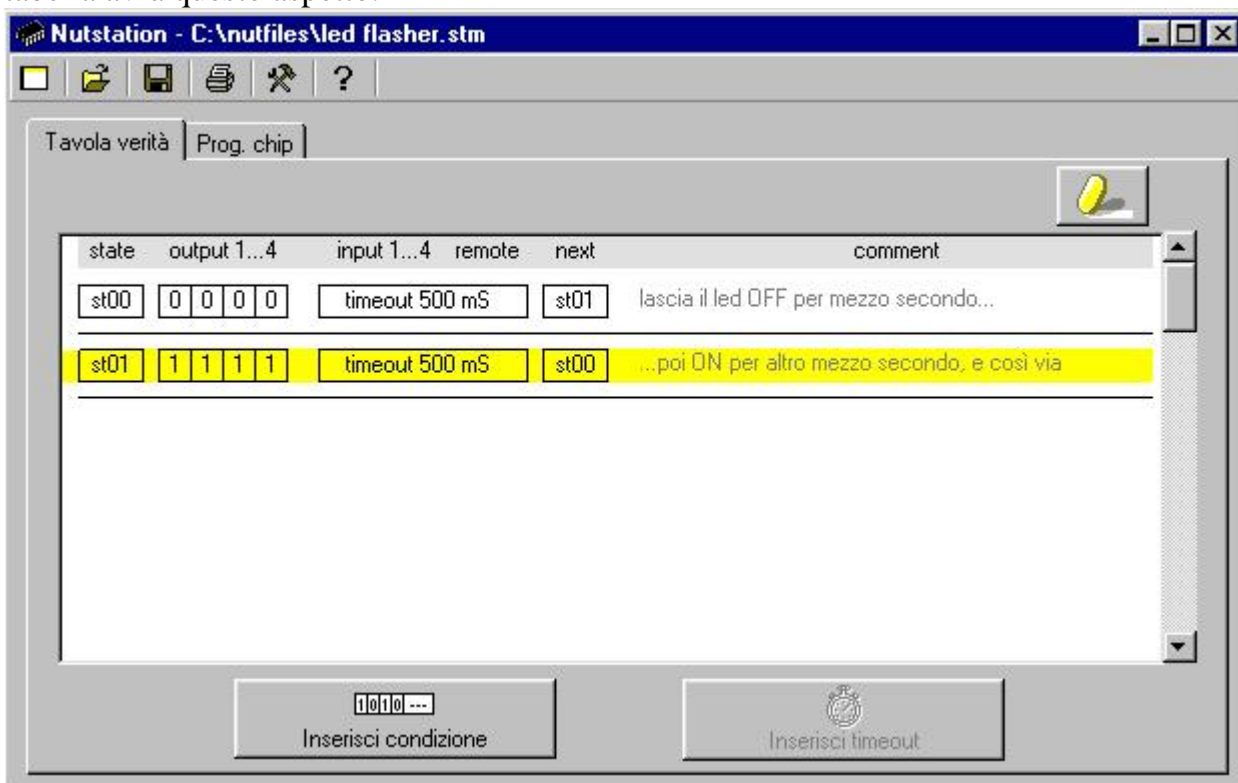
Per schematizzare con carta e penna il funzionamento di un lampeggiatore, possiamo scrivere così:

N. STATO	USCITE	CONDIZIONI E STATO SUCCESSIVO
ST. 0	LED = SPENTO	DOPO MEZZO SECONDO PASSARE ALLO ST. 1
ST. 1	LED = ACCESO	DOPO MEZZO SECONDO PASSARE ALLO ST. 0

La prima colonna identifica lo stato: è solo una numerazione progressiva, che per convenzione comincia da **zero**. Lo **stato zero** è anche lo stato in cui si trova il NUTCHIP al momento dell'accensione, cioè al RESET. Segue la colonna delle **uscite**: infatti il LED è collegato alle uscite. Siamo noi che decidiamo se il LED debba accendersi (uscita OUT4 a livello basso) o spegnersi (uscita OUT4 a livello alto) compilando questa casella. L'ultima colonna riporta le **condizioni**: quando la condizione si verifica, si passa allo **stato successivo** specificato.

In questo esempio è riportata una sola condizione per ogni stato: lo scorrere di un mezzo secondo. Ma i Nutchip permettono di aggiungere anche altre condizioni che tengano conto - oltre che del tempo- anche degli stati logici degli *ingressi* così come dei tasti premuti sul telecomando (*remote*).

Compilieremo ora la tavola della verità utilizzando Nutstation. Quando avremo finito la tabella avrà questo aspetto:




Avrete certo notato una certa somiglianza con la tavola che abbiamo scritto con carta e penna:

-) La prima colonna è il **numero** dello stato (st00, st01, st02, st03...).
-) La seconda colonna raggruppa le **quattro** uscite (**output 1...4**). NOTA: Il LED è collegato all'uscita quattro.
-) La terza colonna raggruppa le condizioni. Le condizioni possono contenere dei temporizzatori (**timeout**, come in questo caso), oppure **combinazioni logiche** di 0 e 1 che devono verificarsi sui 4 ingressi (**input 1...4**). Una particolarità del NUTCHIP

è di essere capace di decodificare i segnali dei telecomandi, per cui si possono inserire condizioni che dipendano dall'aver premuto o meno un tasto del telecomando (**remote**). Vedremo tutte queste possibilità nei prossimi progetti di esempio.

- J) La quarta colonna indica a quale stato si deve passare quando la condizione è vera (**next**). Nel caso di questo esempio è lo stato al quale si passa allo scadere del timeout.
- J) L'ultima colonna è una colonna di **commento**, in cui possiamo una frase a piacere come promemoria e per rendere più semplice l'interpretazione dei vari stati.

Procedura passo passo per creare la tabella degli stati per il lampeggiatore LED:

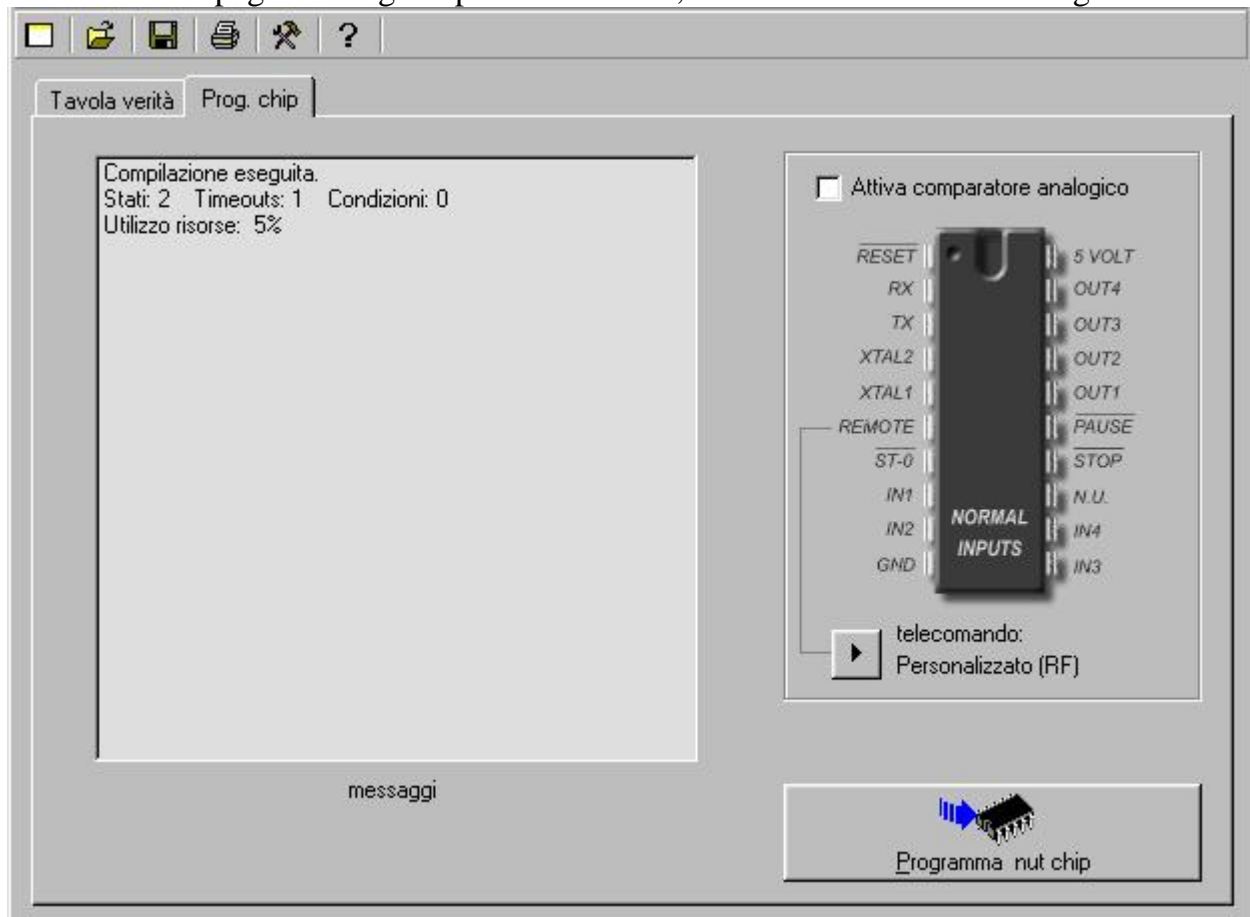
Cliccate su  per cancellare un'eventuale tabella precedente


1. Cliccate sul pulsante "**Inserisci Timeout**" per inserire un timeout (temporizzatore). Si genera un nuovo stato.
2. Lo stato non ha ancora un numero. Al posto del numero compare "???" sulla prima casella. Cliccate sopra la prima casella per impostare "st00".
3. Passate ora alle caselle delle uscite (colonna **outputs 1..4**), e portatele tutte a "1" cliccandoci sopra.
4. Ora accorciate il tempo (colonna **timeout**), che altrimenti rimarrebbe a 1 minuto. Cliccate sul tempo, ed impostate "500 mS" nella finestrella che compare. **NON** evidenziare la casella "rimanente". (Nota: 500mS = 500 millisecondi = 0,5 secondi).
5. Impostate ora lo stato a cui passare allo scadere del tempo (colonna **next**). Anche in questo caso al posto dello stato ci sono dei punti interrogativi, cliccateci sopra fino ad ottenere "st01"
6. Cliccate sul campo del commento (colonna **comment**), e scrivete con la tastiera la frase "LED spento".
La prima riga è ora completa.
7. Cliccate ora in basso su una riga **vuota**: verrà **evidenziata** in giallo.
8. Aggiungete un altro stato cliccando su "**Inserisci Timeout**"
9. Con lo stesso metodo illustrato sopra, impostate lo stato come "st01"
10. Lasciate le uscite tutte a zero, portate invece il tempo (timeout) a 500 mS. Non evidenziate la casella "rimanente".
11. Impostate lo stato successivo (next) a "st00" cliccandoci sopra
12. Battete la frase "LED acceso" nella riga di commento.
13. Controllate il vostro lavoro: dovrebbe essere identico alla figura qui sopra.

Nota: questa tabella è memorizzata nel file "led_flasher.nut" che è fornito assieme a Nutstation.

Programmazione del chip

Portatevi sulla pagina "Prog. chip" di Nutstation, e verificate che sia come segue:



Il riquadro "stato della programmazione" ci informa che non ci sono errori di battitura: se il messaggio nel vostro caso fosse diverso, tornate alla tabella della verità e controllate di non avere commesso degli errori, per esempio inserendo un numero di stato sbagliato. Collegate quindi [l'interfaccia](#) al PC con un cavo seriale. E' il momento giusto per verificare che la porta usata (com1, com2 etc...) corrisponda a quella memorizzata in Nutstation: fatelo cliccando su  (basta farlo una volta sola, in futuro non ci sarà bisogno di ripetere questa operazione).



Infine, inserite l'interfaccia sul circuito e accendete l'alimentazione a 5 volt (attenzione a non invertire la polarità).

Se non sentite puzza di bruciato, potete fare click su "Programma Nutchip"!

In pochi secondi la programmazione è completa: complimenti, il vostro lampeggiatore è pronto!



Alcuni esperimenti

Con i valori indicati il LED lampeggia con cadenza di 1 secondo. Provate a modificarne i tempi: ad esempio, acceso per 1 secondo e spento per 5 secondi; acceso per 0,1 secondi (100 mS) e spento per 0,9 secondi (900 mS).

Provate anche questo valore: acceso per 5 mS e spento per 20 mS: cosa succede? Il LED sembra costantemente acceso, anche se a luce più debole... Ma provate ad allontanarvi di qualche metro e a scuotere rapidamente la testa: vedrete la luce del LED "scomporsi" in segmenti...

Se al posto del led montassimo un **relè**, avremmo realizzato un interruttore ad intermittenza; se montassimo un **buzzer** (cicalino), un semplice avviso sonoro intermittente. Potete dare una sbirciatina alla [scheda riassuntiva](#) dei collegamenti di ingressi e uscite per assaporare il ricco menù offerto dallo chef NUTCHIP !

Il **prossimo passo** sarà quello di utilizzare gli **ingressi** del Nutchip: vedremo come realizzare una **accensione di tipo start-stop** per il LED. L'accensione start-stop è molto utilizzata per le macchine utensili.

In seguito, affronteremo il collegamento di un telecomando a raggi infrarossi, ottenendo un segnalatore LED telecomandato.

[prossimo progetto](#) ►

E se qualcosa andasse storto?

Prima regola: non demoralizzatevi. Ogni errore è una buona occasione per imparare qualcosa di nuovo!

Controllate minuziosamente che tutto corrisponda a quanto descritto fin qui. In particolare:

- J Controllate che il cavo seriale sia effettivamente una prolunga, cioè che il piedino 1 sia collegato con il piedino 1, il 2 con il 2, il 3 con il 3, eccetera (sono in commercio cavi detti "null modem" che NON sono così e quindi non vanno bene).
- J Controllate che la porta seriale del PC sia in ordine e che il suo numero corrisponda a quello memorizzato in Nutstation.
- J Se vi siete costruiti l'interfaccia da soli, controllatela, o meglio sostituirla con una preassemblata oppure fatevene prestare una sicuramente funzionante.
- J Specialmente se la vostra basetta sperimentale non è proprio nuova, assicuratevi che tutti i fili siano ben spelati e restino agganciati saldamente alle molle.
- J Controllate di non avere invertito alcun componente: Nutchip, LED, alimentazione, interfaccia.
- J Ricordate che nelle basette più grandi, le file di contatti esterne di solito sono interrotte a metà
- J Assicuratevi che l'oscillatore ceramico sia effettivamente del tipo a 3 piedini e che sia della frequenza giusta (4 MHz). Il piedino di massa è quello centrale.
- J Controllate con un tester che l'alimentazione sia di 5 volt e che arrivi ai piedini del NUTCHIP .
- J Contate il numero di componenti e di ponticelli di filo che appare sulle fotografie, e controllate che corrisponda a quelli inseriti sul vostro prototipo.
- J Se la programmaizone va a buon termine ma il LED non lampeggia, controllate che i tempi siano in millisecondi (mS) e non secondi (S) o ore (h).
- J Controllate che la tabella della verità corrisponda a quella indocata: eventualmente caricatela dal file originale *led_flasher.nut*.

Infine, se tutti questi suggerimenti non dovessero risolvere il problema, non esitate a chiedere aiuto a un amico più esperto di voi!

Azionamento START-STOP

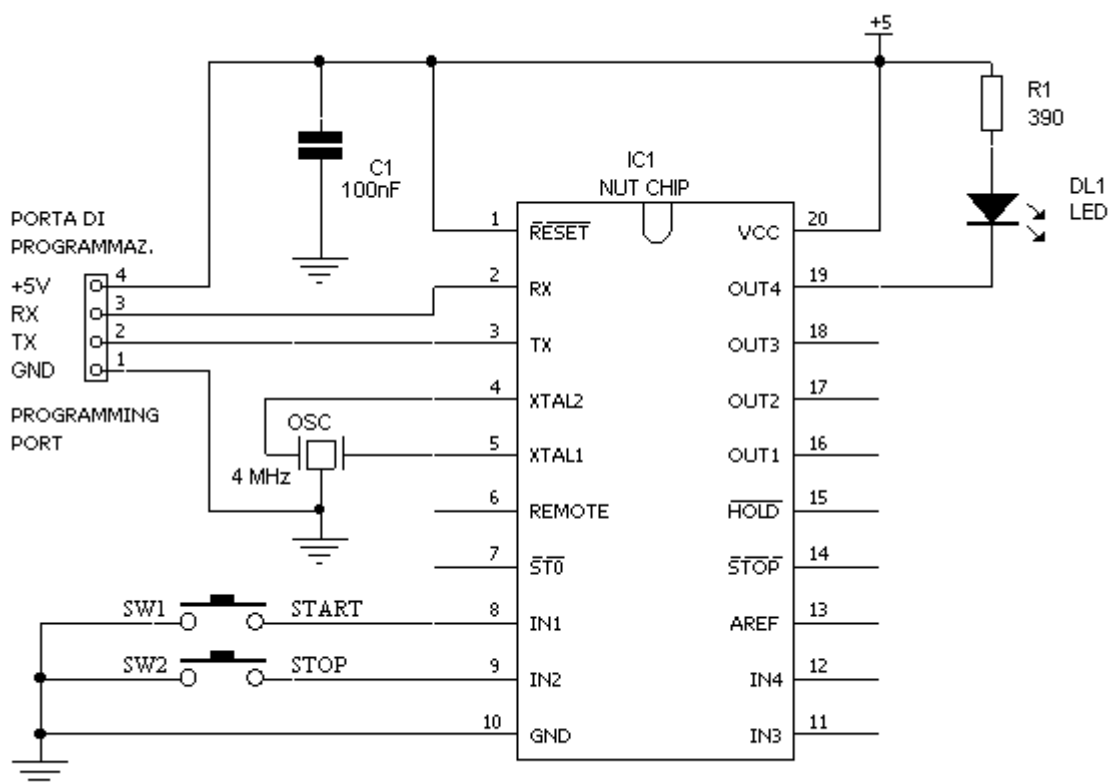
Nella prima parte abbiamo imparato come collegare un LED alle **uscite** e come farlo lampeggiare. In questa seconda parte impariamo come sfruttare gli **ingressi** del Nutchip per fare accendere il LED. Otterremo un'attivazione START-STOP che è quella comunemente usata in tutte le macchine utensili, dove c'è un pulsante per avviare la macchina (START) ed un altro per fermarla (STOP).

Gli ingressi

I Nutchip dispongono di vari **ingressi**. Gli ingressi chiamati **IN1 ... IN4** sono ingressi di uso generale ed il loro funzionamento è obbedisce alla **tavola della verità** che programmiamo nel Nutchip. Un ingresso lasciato **scollegato** è sempre a livello **logico 1**. Collegando l'ingresso alla **massa** (GND) otterremo un livello **logico 0**. Se vogliamo utilizzare un pulsante (o un interruttore), lo collegheremo fra un ingresso e la massa. In questo modo, potremo contare su un **livello 0 quando il pulsante è premuto**, ed un livello logico 1 quando il pulsante è a riposo. Gli ingressi sono importanti perchè possono essere *letti* dai Nutchips: ogni stato dispone di una propria tabella della verità che dice in che stato portarsi in corrispondenza di una determinata combinazione degli ingressi.

Schema elettrico

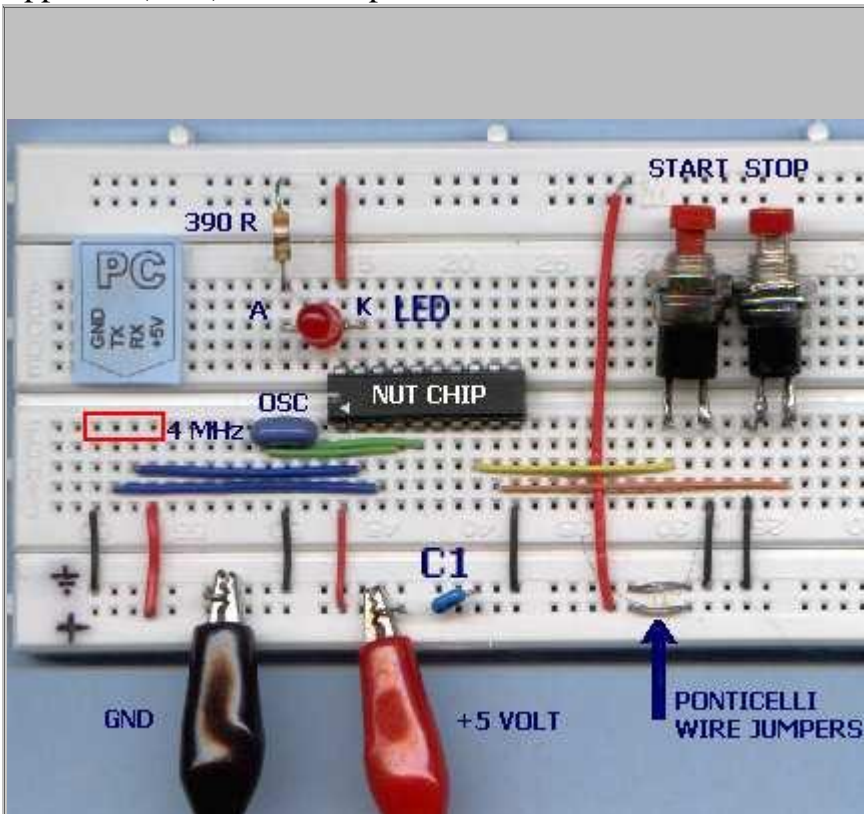
Lo schema è identico a quello del lampeggiatore LED visto in precedenza, a cui aggiungiamo due pulsanti sugli ingressi numero 1 e 2. Questo basterà per provare le funzioni del circuito: più avanti, potremo aggiungere oltre al LED anche un relè, ottenendo un azionamento in grado di accendere carichi come motori o luci. Per il momento, ecco come si presenta il nostro schema:



Montaggio

Aggiungeremo i 2 pulsanti (del tipo con contatto normalmente aperto) alla basetta preparata per la parte 1. Dovremo saldare un pezzettino di filo rigido ai terminali dei pulsanti, che sono troppo grossi per inserirsi nei fori della basetta. Se non avete i pulsanti a portata di

mano, potete *simulare* il contatto con un con uno spezzone di filo rigido. Quando vorremo simulare la pressione del pulsante START o STOP collegheremo a massa il piedino 8 (IN1) oppure 9 (IN 2) del Nutchip.



Elenco dei componenti

-) 1 diodo LED
-) 1 resistenza da 390 ohm
-) 1 risuonatore ceramico da 4 MHz a tre piedini
-) 1 condensatore da 100 nF
-) 2 pulsanti normalmente aperti
-) 1 Nutchip (NUT01-AK)

Occorrono inoltre: un alimentatore a 5V, una basetta sperimentale con contatti a molla, [l'interfaccia Nutchip](#) completa di cavo, vari spezzoni di filo rigido sottile di vari colori, e naturalmente un PC con il [software Nutstation](#).

L'azionamento START-STOP sulla basetta sperimentale

Notate che le piste *orizzontali* della mia basetta sono divise a metà: due ponticelli in filo collegano la parte sinistra con la destra. Verificate che questa interruzione non sia presente anche sul vostro prototipo, altrimenti i pulsanti sembreranno non funzionare!

La tavola della verità

Una attivazione START-STOP si può trovare in due stati:

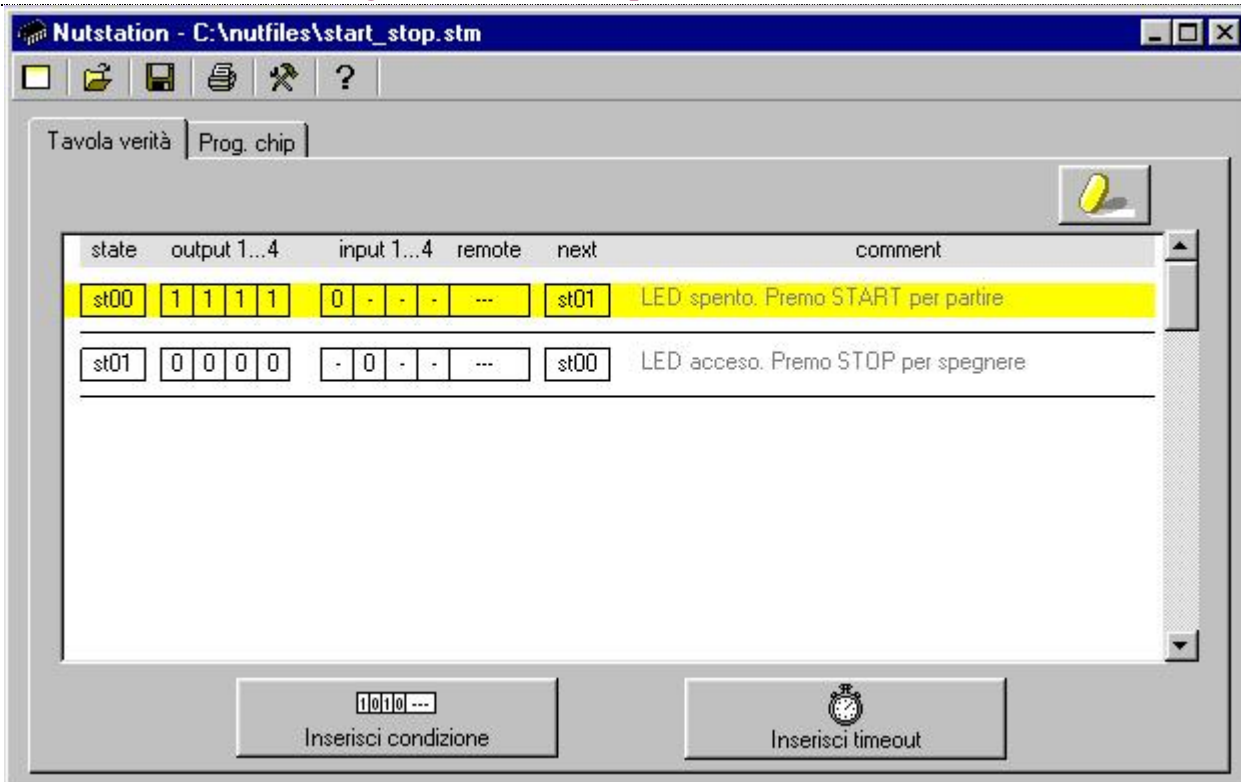
-) può essere **spenta**, ed allora premendo il pulsante START la si accende,
-) oppure può essere **accesa** e premendo il pulsante STOP la si spegne.

Per schematizzarne con carta e penna il funzionamento, possiamo scrivere così:

N. STATO	USCITE	CONDIZIONI	STATO SUCCESSIVO
ST. 0	LED = SPENTO (cioè OUT4 = 1)	SE START E' PREMUTO (cioè IN1 = 0)	PASSARE ALLO ST. 1
ST. 1	LED = ACCESO (cioè OUT4 = 0)	SE STOP E' PREMUTO (cioè IN2 = 0)	PASSARE ALLO ST. 0

Parlando in termini di zeri e uni, che sono gli elementi di ogni tabella della verità, possiamo dire che quando il stato **START** è premuto, l'**ingresso 1** vale **zero**; quando il tasto **STOP** è premuto, è l'**ingresso 2** a valere **zero**.

Per il LED vale la relazione già vista per cui il **LED acceso** con l'uscita 4 a **zero**, e spenta viceversa. Ricopieremo ora la tavola degli stati utilizzando Nutstation. Quando avremo finito la tabella avrà questo aspetto:



Notate la presenza dei commenti nell'apposito spazio: ci aiutano a capire meglio come funziona il progetto, e sono preziosi quando si vuole eseguire una modifica o se si vuole spiegare il funzionamento ad un'altra persona. Ricordiamo che lo stato **zero** è anche lo stato in cui si trova il Nutchip al momento dell'accensione, cioè al **RESET**.

Note: questa tabella è memorizzata nel file "start_stop.nut" che è fornito assieme a Nutstation.

Programmazione del chip

Portatevi sulla pagina "Prog. chip" di Nutstation, e verificate nella finestra di "stato della programmazione" non vengano segnalati errori: se così non fosse, tornate sulla tavola della verità e controllate di non avere commesso sviste, per esempio inserendo un numero di stato sbagliato. Collegate quindi [l'interfaccia](#) al PC con un cavo seriale.



Se non l'avete già fatto, inserite l'interfaccia sul circuito e accendete l'alimentazione a 5 volt (attenzione a non invertire la polarità). Controllate che [l'interfaccia](#) sia regolarmente collegata al PC con un cavo seriale, infine fate click su "Programma Nutchip". In **pochi secondi** la programmazione è completa e potete provare il vostro azionamento START-STOP.



Circuiti derivati

La caratteristica più divertente dei Nutchips è potere modificare la tavola della verità in mille e mille modi diversi, sperimentando altri mille modi di funzionamento che danno luogo ad nuovi utili circuiti. Nel [progetto precedente](#) abbiamo visto come si fa ad inserire un timer nella tavola della verità: ecco allora un **azionamento START STOP temporizzato**. Quando si preme START, il LED si accende per spegnersi dopo 10 secondi - a meno che non si preme STOP prima. Lo trovate nel file: *start_10_stop.nut*:

state	output 1...4	input 1...4	remote	next	comment
st00	1 1 1 1	0 - - - ...		st01	LED spento. Premo START per partire
st01	0 0 0 0	- 0 - - - ... timeout 10 sec		st00 st00	LED acceso. Premo STOP per spegnere oppure aspetto 10 secondi poi spengo

E che dire di un azionamento **misto continuo/intermittente**? Quando si preme START, il LED si accende e resta acceso per 10 secondi, dopo di che comincia a lampeggiare. Questo tipo di funzionamento si utilizza con i *riscaldatori*: all'accensione, il riscaldatore si accende per un tempo determinato per portarsi in temperatura. In seguito, basterà una accensione intermittente per mantenere la temperatura di funzionamento invariata. Troverete la tavola della verità relativa nel file *start_flash_stop.nut*.

state	output 1...4	input 1...4	remote	next	comment
st00	1 1 1 1	0 - - - ...		st01	LED spento. Premo START per partire
st01	0 0 0 0	- 0 - - - ... timeout 10 sec		st00 st02	LED acceso. Premo STOP per spegnere oppure aspetto 10 secondi poi vado a intermittenza
st02	1 1 1 1	- 0 - - - ... timeout 500 mS		st00 st03	Lampeggio (spento). Premo STOP per spegnere aspetto la prossima fase di lampeggio
st03	0 0 0 0	- 0 - - - ... timeout 500 mS		st00 st02	Lampeggio (acceso). Premo STOP per spegnere aspetto la prossima fase di lampeggio

Con questa tavola della verità, quando il LED lampeggia il pulsante START non ha effetto: **sapreste scoprire il perchè?** La soluzione si trova nel file *start_flash_stop_2.nut*:

state	output 1...4	input 1...4	remote	next	comment
st00	1 1 1 1	0 - - - ...		st01	LED spento. Premo START per partire
st01	0 0 0 0	- 0 - - - ... timeout 10 sec		st00 st02	LED acceso. Premo STOP per spegnere oppure aspetto 10 secondi poi vado a intermittenza
st02	1 1 1 1	- 0 - - - ... 0 - - - - ... timeout 500 mS		st00 st00 st03	Lampeggio (spento). Premo STOP per spegnere Premo START per ripartire aspetto la prossima fase di lampeggio
st03	0 0 0 0	- 0 - - - ... 0 - - - - ... timeout 500 mS		st00 st00 st02	Lampeggio (acceso). Premo STOP per spegnere Premo START per ripartire aspetto la prossima fase di lampeggio

Il telecomando

Grazie all'evoluzione tecnologica, oggi possiamo inserire un telecomando anche nei progetti più semplici.

Per utilizzare un telecomando è necessario aggiungere al circuito un ricevitore - oltre naturalmente al telecomando stesso.

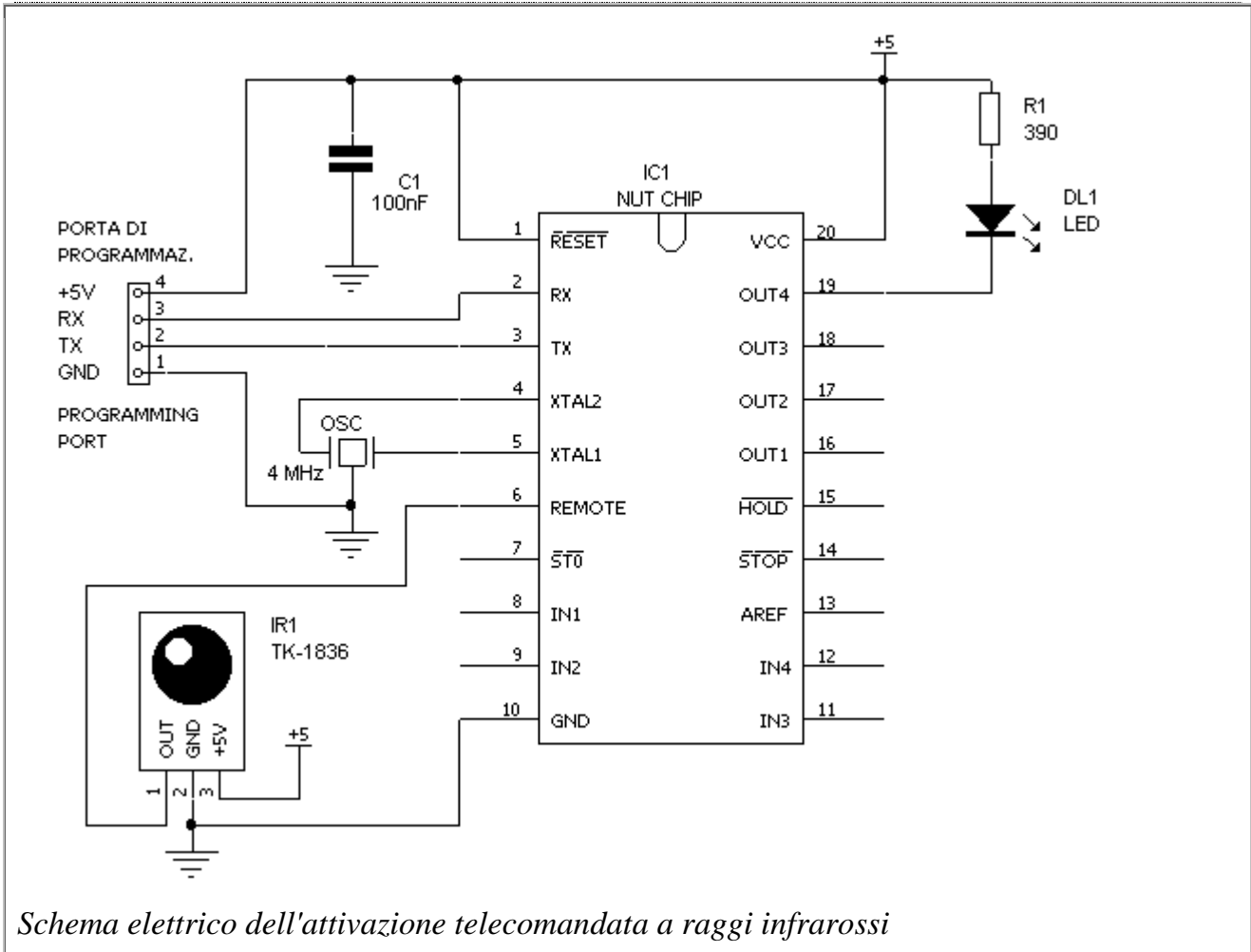
Il **telecomando** emette i segnali codificati: ogni volta che premiamo un tasto (*key*), il codice corrispondente (*code*) viene trasmesso **più volte**. Viaggiando attraverso l'etere o su un fascio di luce infrarossa, i segnali emessi vanno a colpire il **ricevitore**: qui vengono captati e convertiti in impulsi logici a 0...5 volt. Questi impulsi rispecchiano il codice di partenza, ma non sono ancora utilizzabili: infatti vanno interpretati dal **decodificatore** permettono di determinare il pulsante premuto in origine

(1)	ad ogni PULSANTE	il CODIFICATORE all'interno del telecomando	assegna un CODICE
(2)	ogni CODICE viene trasformato	dal TRASMETTITORE all'interno del telecomando	in ONDE RADIO o RAGGI INFRAROSSI
(3)	captando le ONDE o i RAGGI	il RICEVITORE sul circuito	ricostruisce il CODICE
(4)	per ogni CODICE	il DECODIFICATORE all'interno del Nutchip	identifica il PULSANTE premuto in origine

- J **I telecomandi** sono oggi molto diffusi, e si trovano in commercio già pronti come pezzi di ricambio. Sono di due tipi: a radiofrequenza (utilizzati ad esempio negli apricancelli e negli antifurti) e a raggi infrarossi (TV, HiFi e videoregistratori). Questi ultimi si trovano anche presso gli ipermercati.
- J **I ricevitori** sono particolari circuiti integrati o micromoduli che necessitano soltanto dell'alimentazione a 5 volt per potere funzionare. Dispongono di uno o più piedini di uscita dove prelevare il segnale ricevuto, pronto per la decodifica.
- J **Il decodificatore** è incorporato nel Nutchip, ed è collegato al piedino REMOTE. Riconosce fino a 6 pulsanti diversi, con codici a piacere. Questi pulsanti, chiamati *key1...key8*, si possono inserire nella tavola della verità come se fossero dei pulsanti collegati realmente agli ingressi del chip.

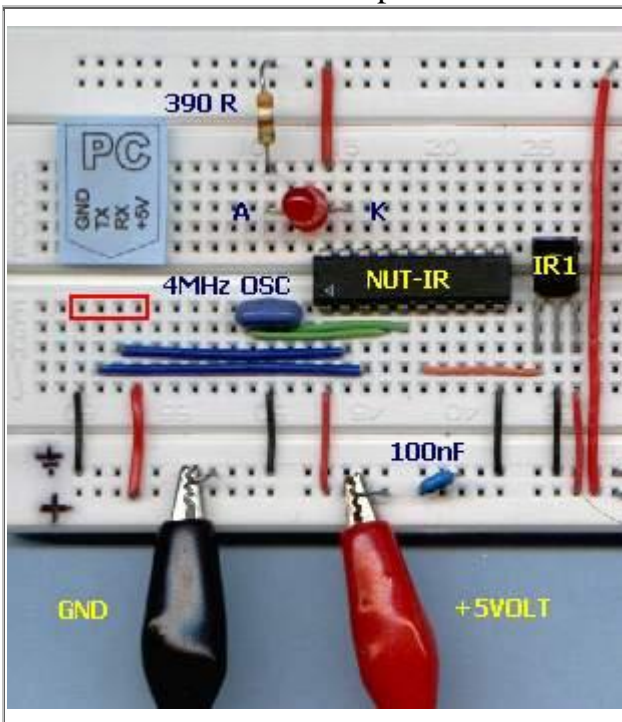
Schema elettrico

Ancora una volta partiamo dallo schema del lampeggiatore LED visto in precedenza, a cui aggiungiamo il ricevitore per raggi infrarossi. Si tratta di un circuito integrato di produzione Temic/Vishay, il TSOP1836 o TK1836. I tre piedini sono l'alimentazione a 5 volt, la massa, e l'uscita di tipo open-collector. Questa uscita richiederebbe una resistenza da 10 kohm verso il positivo, che nel caso dei Nutchip è già incorporata nell'ingresso REMOTE.



Montaggio

Aggiungeremo il ricevitore a infrarossi alla basetta preparata per la [parte 1](#). L'unica precauzione è nell'inserire il ricevitore a infrarossi. Il TK1836 (equivalente al TSOP1836) si presenta come una specie di transistor coi bordi piatti, da cui frontalmente sporge la lente emisferica che è anche la parte sensibile del ricevitore.



Elenco dei componenti

- ⌋ 1 diodo LED
- ⌋ 1 resistenza da 390 ohm
- ⌋ 1 risonatore ceramico da 4 MHz a tre piedini
- ⌋ 1 condensatore da 100 nF
- ⌋ ricevitore infrarossi TK1836 o TSOP1836(Temic)
- ⌋ 1 Nutchip

Occorrono inoltre: un telecomando universale (consigliato il Visa Simplex programmato per il "Gruppo Philips"), un alimentatore a 5V, una basetta sperimentale con contatti a molla, l'interfaccia Nutchip completa di cavo, vari spezzoni di filo rigido sottile di vari colori, e naturalmente un PC con il software

Nutstation.

Il ricevitore infrarossi IR1 può essere sostituito con altri componenti presenti sul mercato. In tal caso utilizzate esclusivamente componenti per telecomandi tipo RC5 con portante a 36kHz.

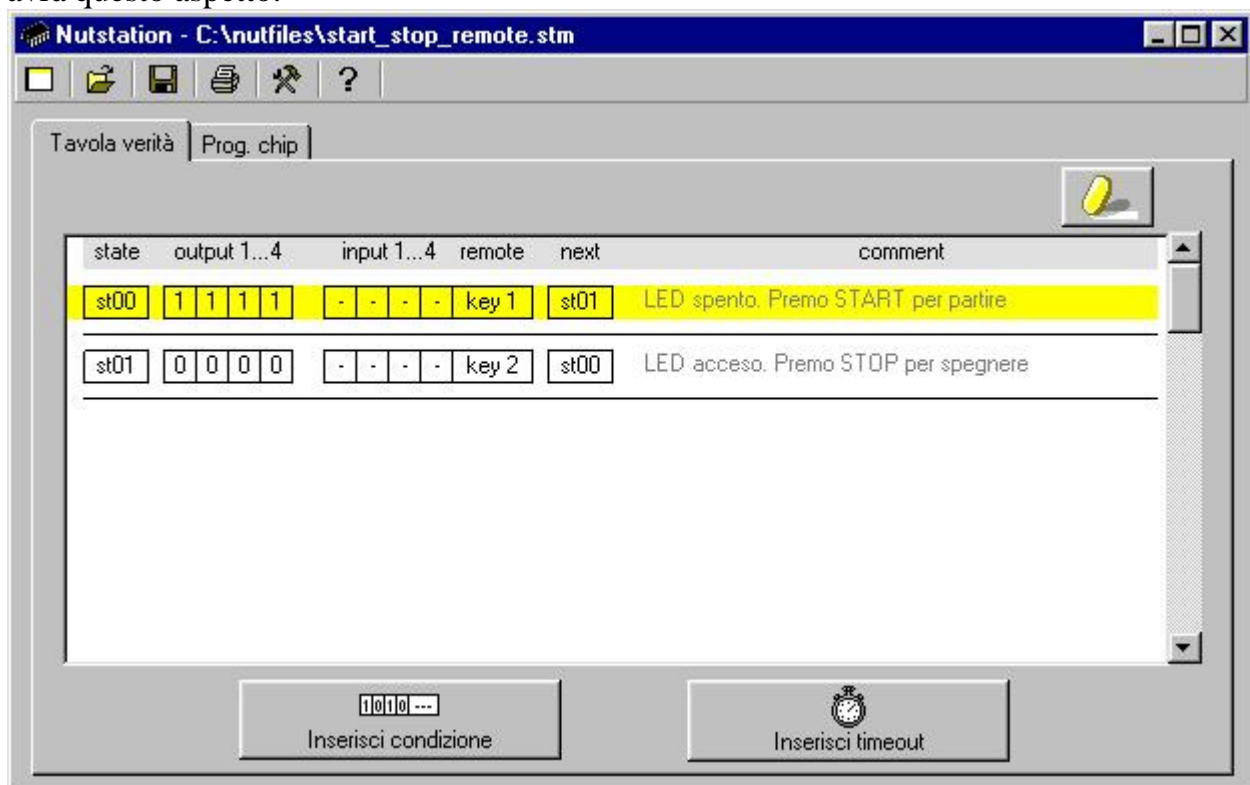
La tavola della verità

I tasti del telecomando si inseriscono direttamente nella tavola della verità, nella casella "REMOTE". Cliccando su questa casella si sceglie un tasto del telecomando, da KEY1 fino a KEY8. Se per esempio scegliamo "KEY1", il NUTCHIP passerà allo stato successivo (NEXT) appena premeremo il pulsante 1 sul telecomando.

La tabella della verità assomiglia molto alla tabella che abbiamo utilizzato nell'attivazione START/STOP. Assumeremo che il pulsante 1 (KEY1) faccia da START, ed il pulsante 2 (KEY2) faccia da STOP.

N. Stato	Uscite	Condizioni	Stato Successivo
ST. 0	LED = SPENTO (cioè $OUT4 = 1$)	SE KEY1 E' PREMUTO	PASSARE ALLO ST. 1
ST. 1	LED = ACCESO (cioè $OUT4 = 0$)	SE KEY2 E' PREMUTO	PASSARE ALLO ST. 0

Ricopieremo ora la tavola degli stati utilizzando Nutstation. Quando avremo finito la tabella avrà questo aspetto:



Notate la presenza dei commenti nell'apposito spazio: ci aiutano a capire meglio come funziona il progetto, e sono preziosi quando si vuole eseguire una modifica o se si vuole spiegare il funzionamento ad un'altra persona. Ricordiamo che lo stato **zero** è anche lo stato in cui si trova il Nutchip al momento dell'accensione, cioè al **RESET**.

***Nota:** questa tabella è memorizzata nel file "start_stop_remote.nut" che è fornito assieme a Nutstation.*

Scelta del telecomando

Abbiamo visto che la tavola della verità identifica i tasti del telecomando numerandoli da KEY1 fino a KEY6. Ma il Nutchip come può sapere qual è veramente il **primo** tasto del telecomando? E qual è il sesto? Molti telecomandi hanno **decine e decine** di tasti...

La soluzione più semplice è utilizzare un telecomando universale già preimpostato: uno di questi è il telecomando ad infrarossi Visa Simplex (tm), che ha 6 tasti ed essendo economico e compatto è molto diffuso. Cliccando invece su "codici infrarossi TV" si inseriscono automaticamente i codici corrispondenti ai pulsanti dei canali da 1 a 6 utilizzati da molti telecomandi TV.



Dalla pagina "Programmazione chip" fate click sul pulsante con la freccina per cambiare il tipo di telecomando usato. Il tipo di telecomando ed i codici che scegliete restano memorizzati automaticamente, perciò se usate sempre il medesimo telecomando non avrete bisogno di selezionarlo ogni volta.

Quando cliccate su un tipo di telecomando ne compare la fotografia che indica le corrispondenze dei tasti da key1 a key6. Potete scegliere uno dei telecomandi standard, cliccando su:

-) codici infrarossi TV (pulsanti dei canali)
-) codici infrarossi Simplex (preprogrammato come "Gruppo Philips")
-) codici RF base

Alternativamente potete personalizzare i codici di un telecomando infrarossi o RF a piacere, cliccando su:

-) infrarossi personalizzato
-) via radio personalizzato



Il Nutchip decodifica gran parte dei telecomandi standard con portante a 36 kHz.

I costruttori TV che adottano tale codifica sono tanti e sarebbe impossibile elencarli tutti. Nutstation dispone di però di una funzione di **autoapprendimento**, che riconosce automaticamente i codici dei telecomandi compatibili (dettagli qui).

Programmazione del chip

Dovreste ormai avere familiarizzato con la programmazione del chip. Portatevi quindi sulla pagina "Prog. chip" di Nutstation, e verificate nella finestra di "stato della programmazione" non vengano segnalati errori: se così non fosse, tornate sulla tavola della verità e controllate di non avere commesso sviste, per esempio inserendo un numero di stato sbagliato.



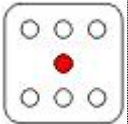
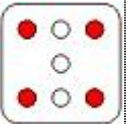
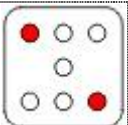
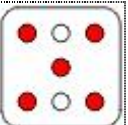
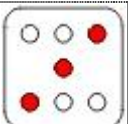
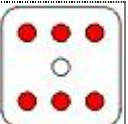
Se non l'avete già fatto, inserite l'interfaccia sul circuito e accendete l'alimentazione a 5 volt (attenzione a non invertire la polarità). Controllate che l'interfaccia sia regolarmente collegata al PC con un cavo seriale, infine fate click su "Programma Nutchip". In **pochi secondi** la programmazione è completa e potete provare il vostro azionamento telecomandato. Premendo il tasto associato a KEY1 il LED si accende. Premendo quello associato a KEY2 si spegne.



Un dado elettronico *utilizza l'ingresso -HOLD del Nutchip*

A prima vista sembrerebbe impossibile realizzare un dado elettronico con un solo Nutchip: un buon dado, per essere realistico, deve utilizzare ben 7 LED: come fare se il Nutchip dispone solo di 4 uscite?

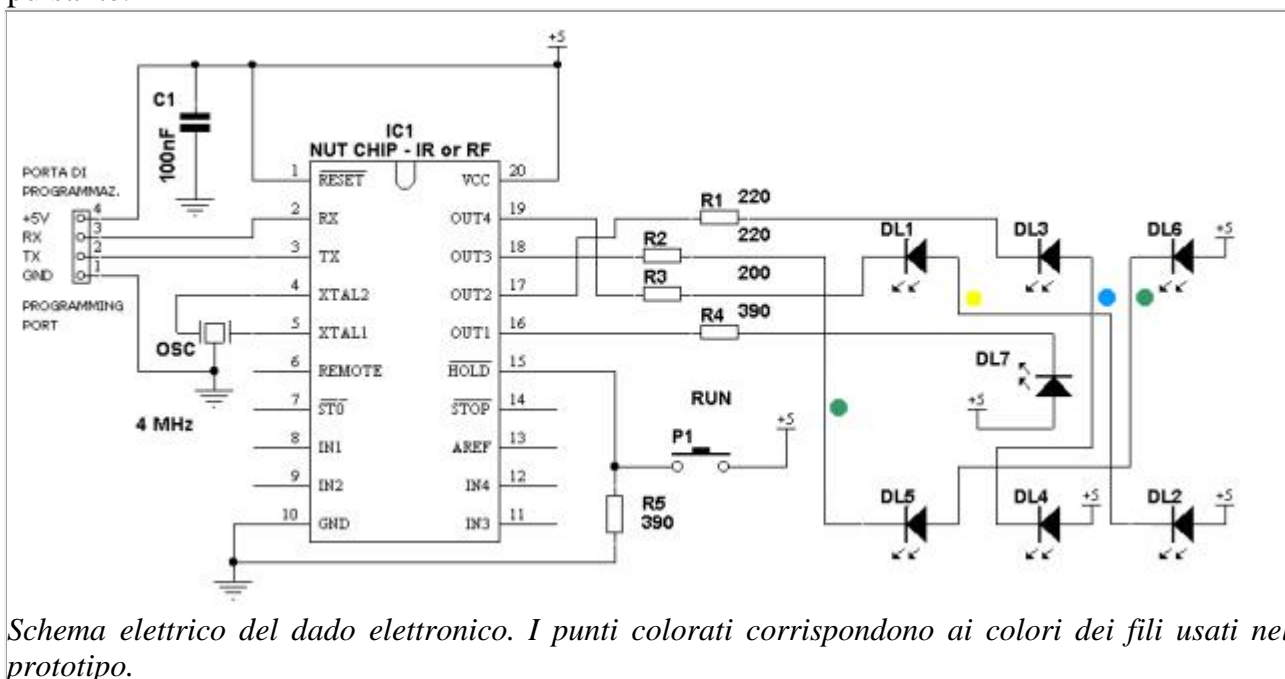
Questo circuito dimostra come 4 uscite siano sufficienti per moltissime applicazioni! Dopo un'analisi un pò meno superficiale, ci accorgiamo che i LED possono essere divisi in 4 gruppi in modo da ottenere tutte le combinazioni di un dado.

	Il numero 1 si ottiene accendendo il solo led DL7		Il numero 4 si ottiene accendendo le coppie di led DL1-DL2 e DL5-DL6
	Il numero 2 si ottiene accendendo la coppia di led DL1-DL2		Il numero 5 si ottiene accendendo le coppie di led DL1-DL2 e DL5-DL6, più il led DL7
	Il numero 3 si ottiene accendendo la coppia di led DL5-DL6 più il led DL7		Il numero 6 si ottiene accendendo le tre coppie di led DL1-DL2, DL3-DL4, DL5-DL6

I quattro gruppi perciò sono rappresentati dalle coppie DL1-DL2, DL3-DL4, DL5-DL6, più il LED singolo DL7. Giusti giusti per le quattro uscite di un Nutchip .

Schema elettrico

Il dado funziona premendo il pulsante P1; i LED lampeggiano freneticamente, per poi arrestarsi quando rilasciamo il pulsante. In questo progetto facciamo uso dell'ingresso -HOLD, che quando viene messo a massa blocca lo scorrere del tempo e quindi ferma il "conteggio" del Nutchip nello stato in cui si trova. Per questo specifico progetto avremmo potuto impiegare anche l'ingresso -STOP indifferentemente. La resistenza R5 da 390 ohm mantiene il piedino di -HOLD ancorato a massa, per cui il NUTCHIP è bloccato ed i numeri non scorrono mostrando l'ultimo numero estratto. Premendo P1 però il piedino si ritrova collegato ai +5V, ed i numeri riprendono a scorrere fino a che non rilasciamo nuovamente il pulsante.



Schema elettrico del dado elettronico. I punti colorati corrispondono ai colori dei fili usati nel prototipo.

A prima vista lo schema dei LED può apparire complicato, perchè nel disegnarlo abbiamo cercato di mantenere le posizioni dei LED e delle resistenze uguali a quelle del prototipo.

Nonostante i percorsi "intrecciati", guardando bene ci accorgiamo che tutti i LED sono collegati a coppie con in serie una resistenza da 220 ohm. Fa eccezione del LED centrale DL7; la sua resistenza R4 è da 390 ohm per ottenere una luminosità uguale a quella degli altri LED.

Tavola della verità

La tavola della verità è non fa altro che mostrare le 6 facce del dado una dopo l'altra. L'intervento del giocatore bloccherà il conteggio in una fase a caso, a seconda di quale stato fosse attivo nel momento in cui il NUTCHIP viene bloccato. Il segreto consiste nel "rotolare" da uno stato al successivo sufficientemente in fretta da impedire di barare. Nel nostro caso abbiamo scelto 30 millesimi di secondo per ogni numero, che dà un gradevole effetto di lampeggio durante il funzionamento.

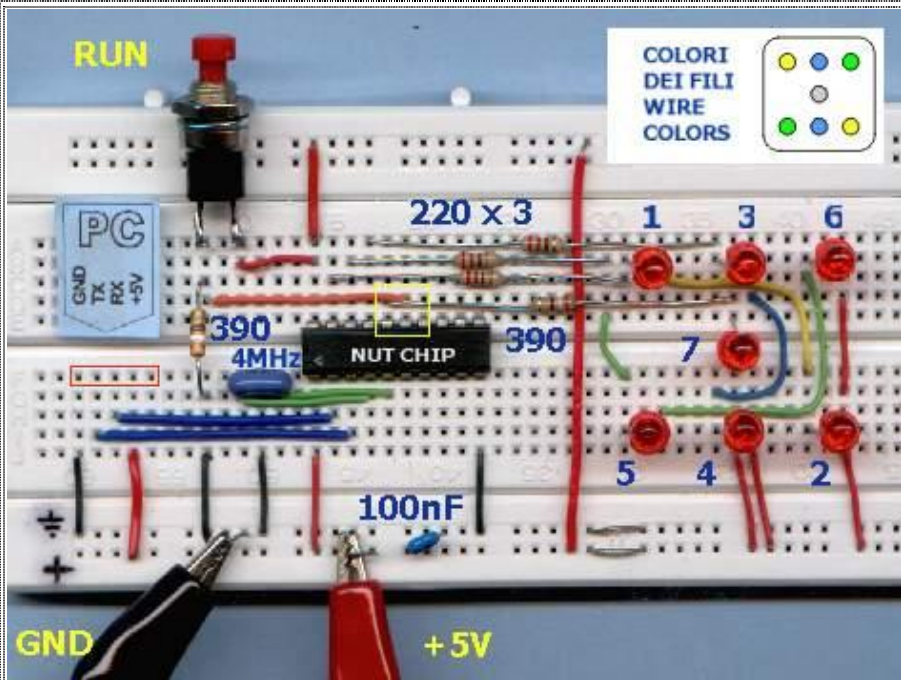
Questa tavola è contenuta nel file *dado.nut*

state	out 1..4	inp 1..4	remote next
st00	0 1 1 1	timeout 30 mS	st01
st01	1 1 1 0	timeout 30 mS	st02
st02	0 1 0 1	timeout 30 mS	st03
st03	1 1 0 0	timeout 30 mS	st04
st04	0 1 0 0	timeout 30 mS	st05
st05	1 0 0 0	timeout 30 mS	st00

Montaggio

Nel montaggio è bene fare molta attenzione nel collegare il LED, perchè è facile confondersi. Tutti i LED sono rivolti con il catodo verso il Nutchip. a parte il led centrale (7) che l'ha rivolto verso l'alto. Nel nostro prototipo abbiamo divaricato i terminali dei LED per lasciare lo spazio di una pista. Lo si vede bene nel LED 2, che ha l'anodo attaccato al filo rosso ed il catodo al filo giallo: in mezzo ai due passa il filo verde.

Per facilitare il montaggio, le resistenze e i LED compaiono nella stessa posizione tanto nello schema così come nel prototipo. Aiutatevi anche con i colori dei fili riportati nelle tabelline. Attenzione anche all'incrocio evidenziato in giallo, fra il filo arancio che va al pulsante e la resistenza da 390 ohm. Questi due fili non devono mai toccarsi fra loro.



Elenco dei componenti:

DL1-DL7: led rossi

R1, R2, R3: 220 ohm

R4,R5: 390 ohm

P1: pulsante normalmente aperto

OSC: oscillatore ceramico a 3 pin, 4MHz

C1: condensatore 100 nF

IC1: NUT01-AK (Nutchip)

Servono inoltre un alimentatore da 5 volt, un'interfaccia PC per programmare il Nutchip ed il software Nutstation

Per il collaudo finale (e se volete capire bene come funziona) aumentate i tempi della tavola della verità portandoli a 2 - 3 secondi. Programmate il chip inserendo l'interfaccia PC nella posizione evidenziata dal riquadro rosso. Premendo il pulsante P1 dovremo vedere scorrere i numeri uno dopo l'altro. Ricaricate poi la tavola con i tempi rimessi a 30 mS per il funzionamento normale.

Un baro elettronico...

Con la tavola della verità proposta, tutte le facce del dado hanno la stessa probabilità di uscire. Ma se aumentiamo il tempo di una delle facce (ad esempio raddoppiandolo) allora raddoppieremo anche la probabilità che quel numero esca! La versione elettronica del dado permette anche una nuova forma di raggio: cioè aumentare le probabilità di due o addirittura tre cifre rispetto alle rimanenti, un artificio difficilissimo da scoprire. Provate a tenere una tabella dei numeri usciti e a fare la statistica di 100 tiri, magari tracciando un grafico di quante volte sono usciti i vari numeri con un foglio elettronico come Excel.

Lampada per TV ...e non solo per la lampada

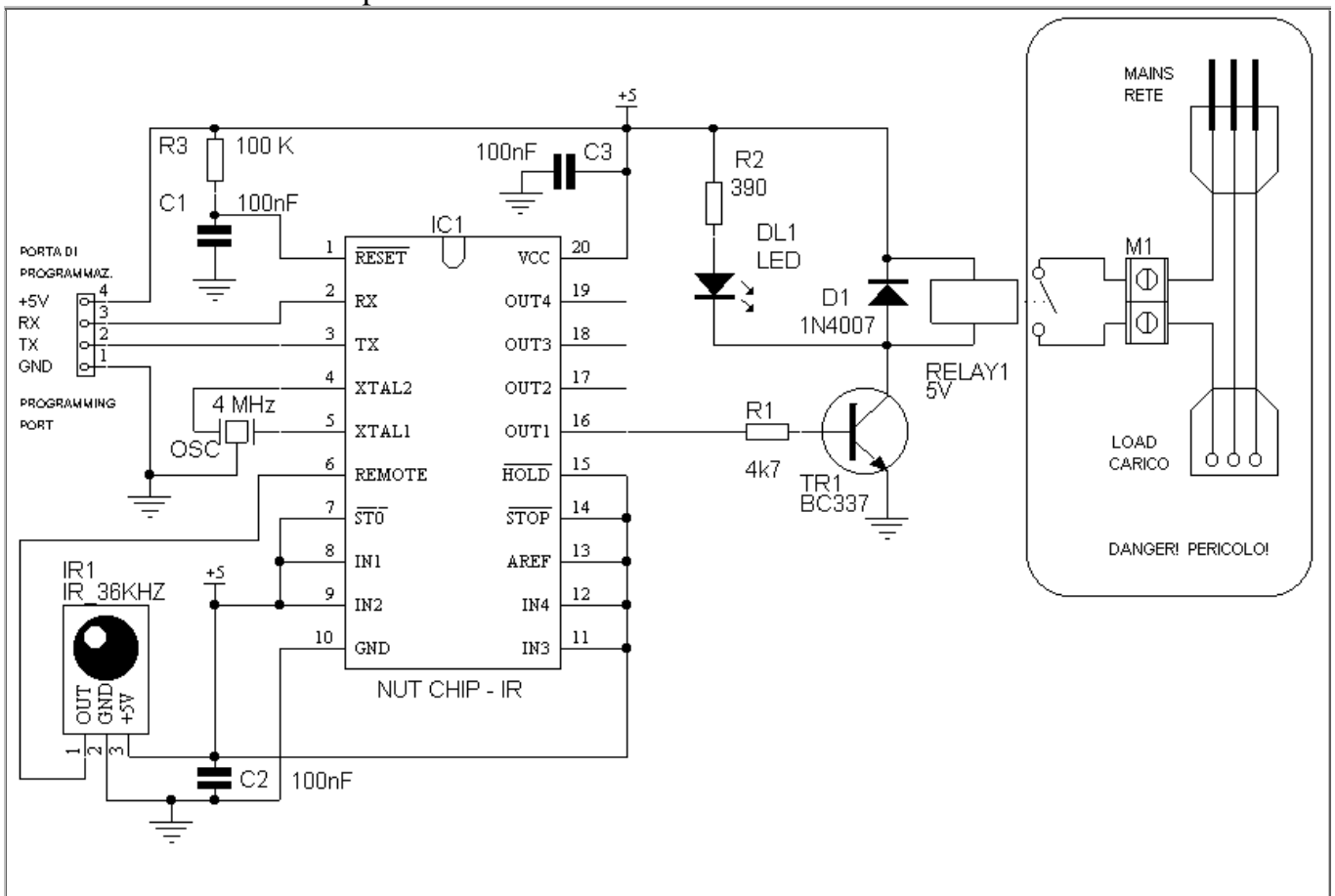
Comodamente in poltrona

La prima applicazione di un telecomando ad infrarossi è per telecomandare una lampada vicino alla TV. Se il telecomando della vostra TV è compatibile, potrete usare un tasto che normalmente non utilizzate (io ho scelto la doppia altezza del televideo). Capita spesso che sul telecomando ci siano dei tasti in più (sono destinati ai modelli "full -optional"), nel qual caso non avete che l'imbarazzo della scelta.

Naturalmente questo è solo uno dei mille usi di questo semplice telecomando ad 1 canale.

Schema elettrico

Dovendo usare il telecomando ad infrarossi, dovremo procurarci oltre al NUTCHIP il modulino ricevitore a infrarossi. Non fatevi ingannare dal nome, perchè più che un modulo questo componente sembra un grosso transistor. Utilizziamo come reset del NUTCHIP una rete RC esterna composta da R3 e C1. I condensatori C2 e C3 invece servono da bypass, e filtrano le alimentazioni del NUTCHIP e del ricevitore infrarossi. Perchè siano efficaci vanno montati molto vicini agli integrati che devono proteggere. Il ricevitore infrarossi è collegato direttamente all'ingresso REMOTE del NUTCHIP ; una resistenza di pull-up non serve perchè è già contenuta all'interno del NUTCHIP . Notate come tutti gli ingressi non utilizzati siano bloccati al positivo.



Il carico (cioè la nostra lampada) è pilotato da un relè. La soluzione più pratica è quella di montare un morsetto sul circuito stampato, e di collegare una spina ed una presa esternamente tramite due cavi lunghi circa un metro. Il circuito viene alimentato a 5 volt (vedi i circuiti base per lo schema di un alimentatore).

E' molto importante che il relè sia in grado di sopportare con un margine adeguato la corrente richiesta dalla lampada. Calcolate gli ampere necessari alla lampada che volete utilizzare ($A = W/V$) e *quadruplicatela*. Ad esempio, per una lampada da 100W, utilizzate

un relè da $4 \cdot 100W/220V = 1,8$ A come *minimo*; in pratica monteremo un modello da 5A. Assicuratevi anche che la corrente e la tensione sopportabili dai contatti sia alternata; alcuni costruttori indicano solo la tensione continua.

Se non riuscite a reperire un relè a 5 volt in grado di sopportare la corrente richiesta, potete ripiegare su uno a 12 volt. In tal caso la resistenza R2 andrà aumentata a 1000 ohm; scollegherete inoltre dai +5 volt sia R2 che D1 (lato catodo) per poi connetterli alla 12 volt.

Non solo il relè, ma anche la spina, la presa ed il cavo utilizzati per collegarsi al carico dovranno essere adeguati: è una buona idea utilizzare una prolunga (reperibile in tutti i supermercati) tagliata a metà.

Montaggio

Poichè si tratta di un circuito che commuta un carico collegato alla rete elettrica, dove le tensioni e le correnti in gioco sono potenzialmente letali, dovremo porre la massima attenzione a ciò che riguarda la sicurezza. E' obbligatorio montare il circuito all'interno di un contenitore plastico completamente isolante. La programmazione del chip ed ogni tipo di misura dovrà essere eseguita con il circuito ed il carico rigorosamente staccati dalla rete elettrica.

Utilizzate un circuito stampato già pronto, a meno che non siate veramente esperti ed allora potete usare anche una basetta "millefori". In tal caso dovrete creare un'area di sicurezza intorno al relè ed al morsetto limando via i "bollini" di rame in modo da lasciare una barriera isolante larga almeno 6 mm attorno ad ogni contatto.

Dopodichè il montaggio non presenta particolari difficoltà: fate attenzione a non invertire la polarità dei componenti ed usate un tester per individuare i contatti del relè prima di montarlo.

Elenco componenti:

- J C1-C2-C3: 100 nF ceramici
- J R1: resistenza 4,7 kohm
- J R2: resistenza 390 ohm
- J R3: resistenza 100 kohm
- J D1: diodo 1N4007
- J DL1: diodo LED rosso
- J OSC: risuonatore ceramico 4MHz 3 piedini
- J RELAY1: relè da circuito stampato, 5V, 10A (vedi testo)
- J M1: morsetto bipolare da circuito stampato, 16A
- J TR1: transistor BC337
- J 1 ricevitore infrarossi 36 kHz (TSOP1836 - TK1836)
- J 1 Nutchip

Varie: prolunga 220V, contenitore plastico, alimentatore 5V

Tavola del Nut

Si tratta di accendere e spegnere l'uscita 1 quando arriva il codice di un tasto: un compito piuttosto semplice per un NUTCHIP .

Per ottenere un funzionamento di tipo on-off premendo due volte un solo tasto, introduciamo un ritardo di un secondo durante il quale il telecomando non viene ascoltato. Ogni volta che premiamo un tasto sul telecomando, infatti, ogni codice viene trasmesso più volte, per qualche decimo di secondo. Il ritardo di un secondo consente al telecomando di "spegnersi". Per capire meglio come funziona, provate a premere il tasto sul telecomando e a non lasciarlo più: vedrete il LED accendersi e spegnersi alla cadenza di un secondo esatto.

state	output 1...4	input 1...4	remote	next	comment
st00	0 0 0 0	- - - -	key 1	st01	luce spenta, si riceve un tasto
st01	1 0 0 0	timeout 1 sec		st02	accendiamo la luce, ignoriamo il tlc per 1 secondo
st02	1 0 0 0	- - - -	key 1	st03	luce accesa, si riceve un tasto
st03	0 0 0 0	timeout 1 sec		st00	spegniamo la luce, ignoriamo il tlc per 1 secondo

Tavola degli stati da inserire nel NUTCHIP . Potete trovarla nel file tv_light.nut

Programmate la tavola nel NUTCHIP , senza dimenticare di impostare il tasto da abbinare a key1. Se come spesso accade il vostro telecomando ha qualche tasto in più che non viene usato, il problema è risolto. Altrimenti potete usare un tasto di una funzione televideo che non usate mai: ad esempio, il tasto "rivela risposte", o "doppia altezza". Altri telecomandi inoltre hanno un pulsante laterale o un interruttore che gli permette di funzionare come il pulsante del videoregistratore: anche qui potrete individuare molti tasti da "riciclare".

Modifiche geniali

Questo telecomando per la sua semplicità si presta a mille piccole modifiche per adattarsi alle esigenze più disparate.

- J **Servono impulsi?** Anziché il funzionamento ON/OFF come proposto in questo progetto, potete modificare la tavola della verità per produrre una chiusura del relè (da uno a due secondi di durata) tutte le volte che si aziona il pulsante. Questo tipo di funzionamento può essere necessario per azionare aperture, antifurti, campanelli, e simili.
- J **Serve un timer?** L'accoppiata telecomando+timer è davvero milleusi, come dimostra l'invenzione di Claudio qui a lato.
- J **Servono pulsanti?** Potete aggiungere due pulsanti (on/off) agli ingressi IN1 e IN2 del NUTCHIP , utili per pilotare il carico localmente (...o quando le batterie del telecomando si scaricano!)
- J **Servono più canali?** Non è difficile aggiungere canali a questo telecomando, noi lo abbiamo fatto e lo descriviamo nel progetto di un completo telecomando a tre canali.
- J **Serve più portata?** Utilizzate la radiofrequenza, collegando al NUTCHIP un ricevitore radio ed un telecomando UHF al posto dei raggi infrarossi, seguendo le direttive dei collegamenti base.
- J **Scatenate le idee...** questo circuito è davvero versatile, potete trasformarlo addirittura in un chiave a combinazione...invisibile!

L'acqua calda riscoperta

La vecchia caldaia di Claudio (Ravenna) proprio non ce la fa a scaldare l'acqua per la doccia assieme a quella del termosifone.

E il geniale Claudio cosa ti fa?

Modifica la tavola del Nut per fare scattare il relè per dieci minuti ogni volta che preme il pulsante. Collega poi il relè in modo da spegnere la pompa di circolazione del termosifone durante la doccia: per dieci minuti tutto il calore si concentra per scaldare l'acqua del bagno, per poi tornare automaticamente ai termosifoni quando il relè torna a riposo.

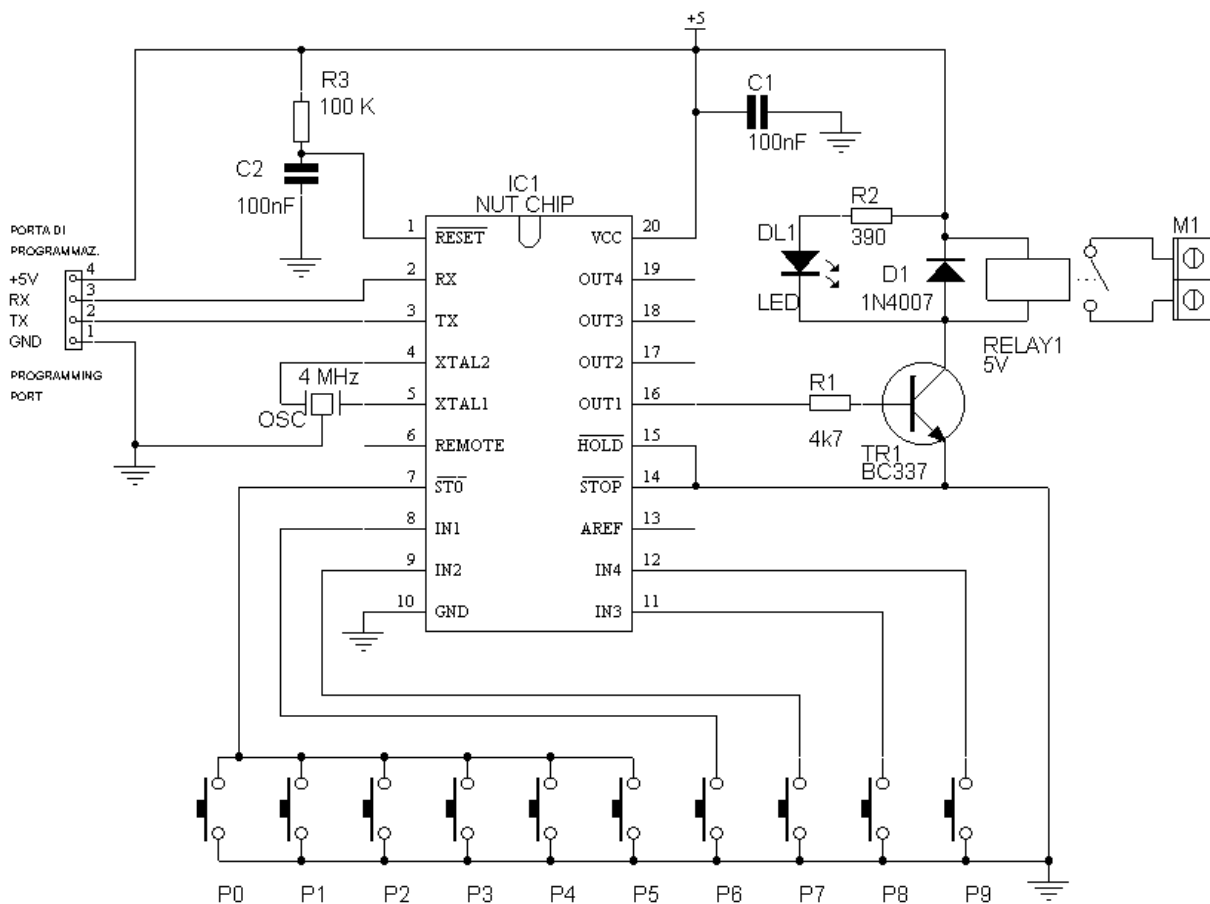
Per attivare il telecomando dal bagno Claudio ha usato un telecomando via radio anziché a infrarossi, collegato come nel progetto a tre canali

Chiave a combinazione utilizziamo l'ingresso st0

Solo una combinazione fra le 9.999 possibili è in grado di aprire questo blocco elettronico. Una chiave a combinazione non è solo un ottimo sistema per proteggersi e per proteggere, è anche un valido deterrente che ammonisce sulla possibile presenza di un instancabile guardiano con un cuore di silicio. Si utilizza direttamente per sostituire un lucchetto ed una chiave tradizionali, specialmente quando più persone sono autorizzate ad entrare: servizi condominiali, ristoranti, spogliatoi, garage comuni, magazzini, negozi sono solo alcuni esempi di luoghi dove l'utilizzo di una chiave tradizionale è perlomeno scomodo: senza considerare che cambiare la combinazione è più semplice che non cambiare una serratura e ridistribuire decine di chiavi... Ma la chiave elettronica fa molto di più: non blocca solo l'accesso ad un locale, ma persino ad una singola macchina. I soliti furbi dovranno fare i conti con la sua logica inflessibile. Eccola allora pronta a sorvegliare una linea telefonica, una fotocopiatrice, un televisore, un fax, un'autoclave. E non abbiamo ancora parlato della possibilità di aggiungere un telecomando...

Schema elettrico

La chiave elettronica è composta da una tastiera, dal Nutchip e da un relè. La tastiera ha dieci tasti, contrassegnati ognuno da una cifra diversa da zero a nove. Di questi pulsanti, quattro sono collegati ai quattro ingressi IN1...IN4 del NUTCHIP. Il Nutchip viene programmato in modo che, per ottenere l'attivazione del relè, è necessario portare a zero prima l'ingresso IN1, poi IN2, poi IN3 ed infine IN4. Se utilizziamo i collegamenti come disegnati sullo schema otterremo la combinazione sarà il numero 6-7-8-9. Qualsiasi altra sequenza blocca la combinazione e bisogna ricominciare daccapo. Per questo gli altri sei tasti sono uniti fra loro e sono collegati al pin -ST0. Questo piedino è un ingresso a logica fissa, e quando viene portato a zero ha l'effetto di trascinare il NUTCHIP allo stato st00, a prescindere dallo stato degli altri piedini ed anche dalla tavola degli stati programmata nel chip.



Ovviamente possiamo scegliere una qualsiasi altra combinazione di quattro cifre: se ad esempio volessimo la combinazione 1-7-8-9 dovremmo semplicemente scambiare di posto P1 con P6. Se invece la tastiera fosse già realizzata, possiamo rinfrescare la combinazione cambiando semplicemente l'ordine di inserimento delle cifre: in tal caso non ci servirà il saldatore ma basterà una ritoccatina alla tavola degli stati.

Tavola degli stati

Si tratta quindi di:

-) richiedere che gli ingressi IN1...IN4 si abbassino uno dopo l'altro: se ciò avviene, si aziona il relè per 5 secondi
-) annullare l'operazione se uno degli ingressi IN1...IN4 si chiude al di fuori della sequenza
-) annullare l'operazione se uno degli altri pulsanti (collegati alla linea -ST0) viene premuto
-) come ulteriore sicurezza, diamo un tempo massimo di 5 secondi per inserire ogni cifra, oltre il quale l'operazione si annulla automaticamente

Alla partenza il relè è disattivato e in attesa di una nuova combinazione. Questo soddisfa anche la terza condizione, in quanto premendo uno dei tasti "non abilitati" che sono collegati all'ingresso speciale -ST0 ci ritroveremo immediatamente in questo stato.

Se invece premiamo uno dei tasti giusti, gradualmente avanziamo da st00 in st01, poi in st02, in st03 ed infine in st04, dove si attiva il relè per cinque secondi per poi ritornare alle condizioni iniziali. In ogni stato è presente un timer da 5 secondi: se entro tale tempo non inseriamo la cifra seguente, la combinazione si annulla e si ricomincia daccapo.

Notate infine che anche i tasti collegati ad IN1...IN4 sono anche controllati singolarmente, stato per stato: non vale premerli prima o dopo che sia venuto il loro turno. Ecco perché se IN4 va a zero mentre ci troviamo nello stato st01 la combinazione viene abbandonata immediatamente, mentre se lo premiamo quando ci troviamo nello stato st03 il relè si sblocca.

state	output 1...4	input 1...4	remote	next
st00	0 0 0 0	0 - - - ... - 0 - - ... - - 0 - ... - - - 0 ...		st01 st00 st00 st00
st01	0 0 0 0	- 0 - - ... - - 0 - ... - - - 0 ...	timeout 5 sec	st02 st00 st00 st00
st02	0 0 0 0	0 - - - ... - - 0 - ... - - - 0 ...	timeout 5 sec	st00 st03 st00 st00
st03	0 0 0 0	0 - - - ... - 0 - - ... - - - 0 ...	timeout 5 sec	st00 st00 st04 st00
st04	1 1 1 1		timeout 5 sec	st00

tavola degli stati del lucchetto elettronico, contenuta nel file keylock.nut

Per gli sperimentatori

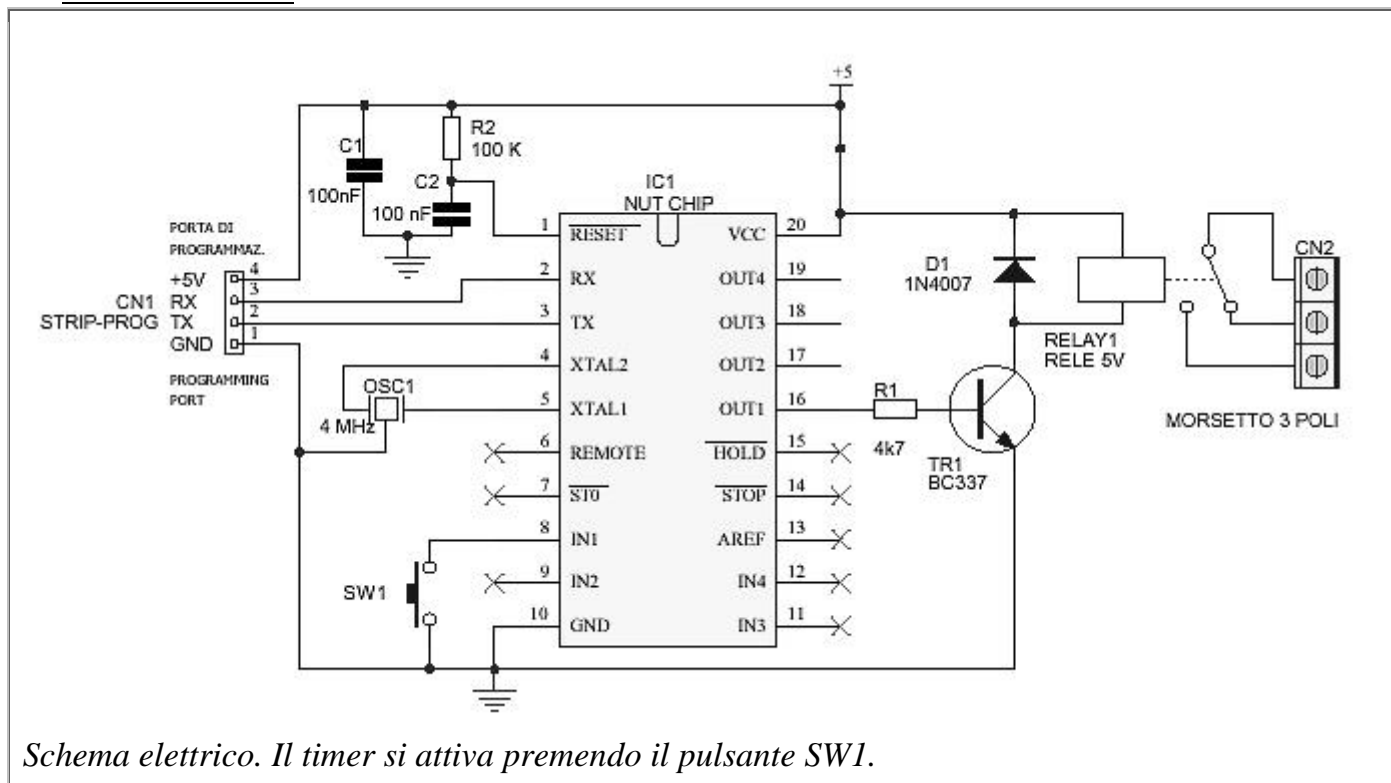
Questo progetto si presta ad essere personalizzato in molti modi. Ecco alcuni stuzzichini per il vostro appetito di sperimentatori:

-) Aggiunta di cifre: nulla vieta di introdurre una combinazione a 6 cifre, ad esempio 6-7-6-8-6-9. Le combinazioni saliranno da 9.999 quasi un milione: esattamente 999.999. Ovviamente dovremo usare le cifre dei pulsanti collegati agli ingressi IN1...IN4 ripetendole più volte. Sapreste farlo?
-) Se siete stati così bravi da portare la combinazione a 6 cifre, sapreste dire perchè la combinazione 66-77-88 (composta ripetendo due volte la stessa cifra) non funziona bene? Sapreste modificare la tavola in modo da utilizzare anche questo tipo di cifre?
-) Aggiunta di un telecomando, per realizzare, ad esempio, il controllo della sbarra di un parcheggio. Chi può, telecomanda. Chi non può, digita!

Collezione di Timer - facili, utili, versatili

Il timer è un circuito fondamentale dell'elettronica e si presta ad una infinità di applicazioni. Ci sono tantissimi tipi di timer in commercio, uno diverso dall'altro. Il Nutchip si presta molto bene alla costruzione di un timer: lo stesso circuito, programmato in modo diverso, copre le funzioni di una grandissima gamma di timer commerciali, con una precisione invidiabile.

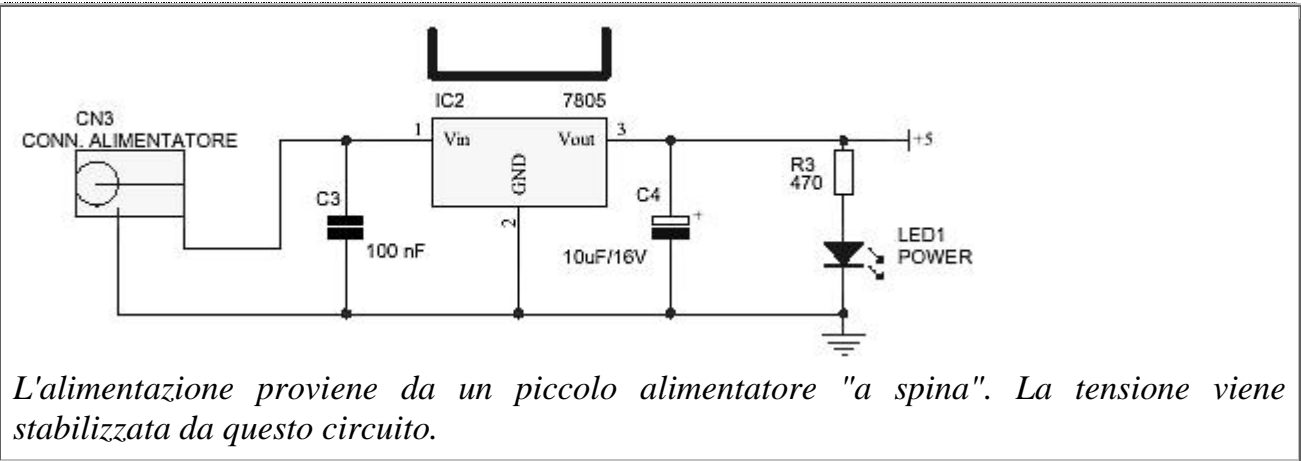
Schema elettrico



Schema elettrico. Il timer si attiva premendo il pulsante SW1.

Lo schema elettrico è molto semplice. Il pulsante SW1 collegato all'ingresso IN1 serve per azzerare il timer e fare **partire** il tempo. Le uscite vanno a livello 1 allo **scadere** del tempo. Il transistor **TR1** - pilotato dall'uscita del Nutchip tramite la resistenza di limitazione R1 - fornisce la potenza necessaria per **eccitare il relè**. Il diodo D1, collegato ai capi della bobina, protegge il transistor dalle tensioni elevate che si generano sull'avvolgimento del relè al momento della commutazione. Quest'ultimo va scelto con **eccitazione** a 5 volt e contatto di uscita con caratteristiche adeguate al carico che si vuole pilotare, tenendo un **marginale di sicurezza** di almeno il 50%. Ad esempio, se vogliamo accendere una lampada alogena a 12 volt che assorbe 3 ampere, sceglieremo un relè il cui contatto di **uscita** sia certificato per almeno 24V e 5A.

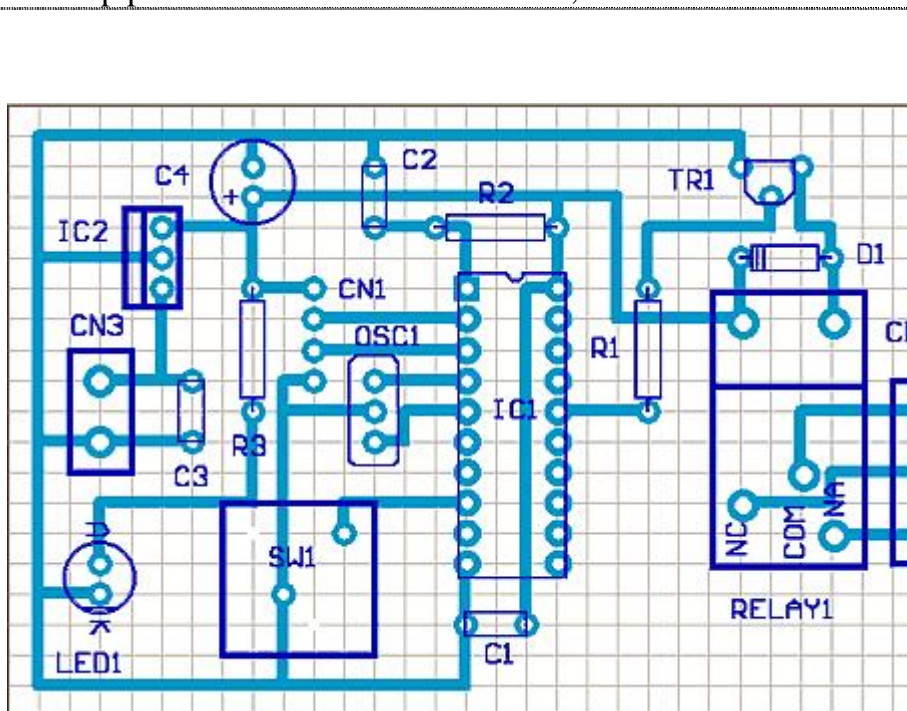
Il Nutchip è illustrato con un oscillatore di tipo ceramico, che ha una precisione migliore dell'1%. Ma se volesse ottenere una precisione ancora **migliore** (per esempio per realizzare timer *giornalieri* o *settimanali*), utilizzate un **quarzo** come indicato nella raccolta dei circuiti base. Nella stessa raccolta troverete dei collegamenti alternativi per il RESET, che qui per semplicità è portato direttamente al positivo.



Montaggio

Il circuito è molto semplice e si presta bene ad essere montato sia sull'apposito circuito stampato, che su di una basetta sperimentale tipo "millefori". Il circuito stampato ospita anche i componenti dell'alimentatore. Cominciate sempre dai componenti più bassi, tenedo per ultimi i componenti più ingombranti che altrimenti ostacolerebbero il lavoro. Non saldate direttamente il Nutchip, ma utilizzate uno zoccolo. Fate estrema attenzione a montare il relè, in modo che i fili e le saldature dei contatti siano bene isolate dal resto del circuito. Un contatto indesiderato fra i fili che vanno al morsetto M1 ed il resto del circuito può risultare in pericolosissime scosse elettriche, oltre alla distruzione del circuito.

Vi consigliamo di usare uno zoccolo a 20 piedini per il Nutchip. Se non è richiesta la programmazione *in-circuit*, cioè senza smontare il Nutchip dalla scheda, potete omettere il connettore CN1 per la porta di programmazione. In tal caso però dovrete programmare il Nutchip prima di collocarlo sulla basetta, utiizzando un'altra scheda collegabile al PC.



Elenco dei componenti

- J R1=4,7 K \varnothing
- J R2=100 k \varnothing
- J R3=470 \varnothing
- J OSC1=1 risuonatore ceramico da 4 MHz a tre piedini (vedi testo)
- J TR1=BC237
- J D1=1N4007
- J RELAY1= relè 5 volt (vedi testo)
- J C1,C2,C3=100 nF
- J C4=10uF/16V
- J SW1 pulsante norm_aperto
- J CN2= morsetto a 2 poli
- J IC1=Nutchip NUT01-AK completo di zoccolo
- J IC2=7805 con aletta
- J Alimentatore 7...12V
- J basetta sperimentale

Piazzamento dei componenti sul circuito stampato. Potete anche servirvi di una basetta "millefori": in tal caso seguite la quadrettatura (che è proprio del passo usato nelle basette sperimentali) unedo le piazzole con dei fili di rame nudo.

tipo "millefori"

Le tavole della verità

Con questo timer possiamo ottenere una grandissimo assortimento di timer differenti, semplicemente utilizzando programmi diversi. Ogni tipo di timer è idoneo per una specifica applicazione, con tempi che vanno dai pochi secondi a molte ore.

Timer monostabile.

Questo timer aziona il relè appena si schiaccia il pulsante, e si stacca dopo il tempo prefissato nello stato st01. Successive pressioni del pulsante all'interno del tempo di attivazione non hanno alcun effetto.

Applicazioni:

ingranditore fotografico, spruzzatori, dosatori di liquidi, avanzamento di un motore di una quantità prefissata, tempo di apertura di un cancello.

state	output 1...4	input 1...4	remote	next	comment
st00	0 0 0 0	0 - - - ...		st01	relè OFF: attendiamo la pressione del pulsante...
st01	1 1 1 1	timeout 1 min		st00	relè ON: temporizzazione in corso!

Il file "timer_monostabile.nut" è impostato per 1 minuto. Potete allungare il tempo fino a mille ore!

Accensione ritardata

Il tempo parte appena si schiaccia il pulsante. Il relè si attiva soltanto allo scadere del tempo e resta acceso finchè non si toglie corrente al circuito.

Applicazioni:

Accensione elettrodomestici, timer da cucina (allo scadere del tempo si aziona un buzzer), tempo massimo in giochi a quiz, accensione antifurto, anti-bump per casse acustiche (cortocircuitare SW1), parchimetri.

state	output 1...4	input 1...4	remote	next	comment
st00	0 0 0 0	0 - - - ...		st01	relè OFF: attendiamo la pressione del pulsante...
st01	0 0 0 0	timeout 1 hour		st02	relè OFF: ritardo di accensione in corso...
st02	1 1 1 1	- - - - ...		st02	relè ON: finchè non si toglie corrente!

*Con questo esempio potete azionare la lavatrice 1 ora dopo essere usciti da casa
Questa matrice è contenuta nel file "timer_ritardato.nut".*

Allungatore di impulso

Aziona il relè appena si schiaccia il pulsante: ripremendolo o tenendolo premuto il tempo ricomincia daccapo. Il relè si disattiva allo scadere del tempo.

Applicazioni:

Timer per luci scale, ventilatori per toilette, timer per pompe, rubinetti e asciugamani elettrici.

state	output 1...4	input 1...4	remote	next	comment
st00	0 0 0 0	0 - - - ...		st01	relè OFF: attendiamo la pressione del pulsante...
st01	1 1 1 1	0 - - - ... timeout 3 min		st01 st00	se si ripreme, rientriamo in st01 e ricomincia il tempo relè OFF: ritardo di accensione in corso...

Un timer perfetto per le luci scale, perchè si "ricarica" finchè si tiene premuto il pulsante

Lo trovate nel file *"timer_allungatore.nut"*

Rivelatore di mancanza di impulso

Bisogna premere il pulsante entro il tempo prefissato, altrimenti il relè scatta.
Ad esempio, se il tempo è regolato su 5 secondi, bisogna premere il pulsante almeno una volta ogni 5 secondi, o più spesso.

Applicazioni:

Macchine automatiche per segnalare la mancanza di un pezzo, per rivelare l'arresto di un meccanismo, per sistemi di sorveglianza (il guardiano deve premere il pulsante periodicamente).

state	output 1...4	input 1...4	remote	next	comment
st00	0 0 0 0	0 - - - timeout 5 sec	---	st00 st01	se si preme il pulsante, si "ricaricano" i 5 secondi non è arrivato l'impulso: accendiamo il relè...
st01	1 1 1 1	- - - -	---	st01	relè ON: finchè non si toglie corrente!

*Se il contatto SW1 resta aperto per più di 5 secondi, il relè si accende.
Se SW1 è collegato ad un nastro, il relè rivelerà la mancanza del movimento.
' il file "timer_rivelatore.nut"*

Timer ciclico

E' un timer a doppio tempo: il primo tempo stabilisce la durata di accensione del relè (tempo di ON) , il secondo tempo la durata di spegnimento (tempo di OFF). Il tempo ON parte appena si schiaccia il pulsante. Successive pressioni fanno ripartire daccapo il tempo di ON.

Applicazioni:

Intermittenze, insegne luminose, decorazioni natalizie, economizzatori di consumo

state	output 1...4	input 1...4	remote	next	comment
st00	0 0 0 0	0 - - - timeout 5 min	---	st00 st01	relè ON
st01	1 1 1 1	0 - - - timeout 5 min	---	st00 st00	relè OFF

Per realizzare un modulatore per riscaldatori da acquario sostituiremo SW1 con un interruttore.

*Finchè l'interruttore resta chiuso, il riscaldatore elettrico resta acceso al 100%.
Aprendo SW1, si accenderà con una cadenza di 5 minuti, riducendo la potenza media al 50%.File: "timer_ciclico.nut"*

Timer giornaliero

E' un timer ciclico in cui la somma del tempo ON e del tempo OFF dà esattamente 24 ore.
Appena si schiaccia il pulsante il relè si attacca, e resta acceso per tutto il tempo specificato nello stato stxx.Si riattaccherà il giorno successivo, esattamente alla

Applicazioni:

Accensione di insegne e vetrine, impianti di riscaldamento, innaffio giardini, illuminazione notturna e anti-ladro.

stessa ora.

state	output 1...4	input 1...4	remote	next	comment
st00	1 1 1 1	0 - - - ... timeout 1 hour		st00 st01	relè ON per 1 ora
st01	0 0 0 0	0 - - - ... timeout 23 hour		st00 st00	pulsante premuto, sincronizziamo gli orologi... altrimenti relè OFF per tutto il resto del giorno

Un timer per innaffio giardini con il sistema a microgocce: si accende per un'ora, tutti i giorni.

E' importante che i tempi ON+OFF diano esattamente 24 ore.

In questo esempio, tratto dal file "timer_giornaliero.nut", il tempo ON=1 ora, il tempo OFF=23 ore.

E' solo l'inizio

Come avete visto, realizzare un timer non è difficile. Il timer è un punto di partenza per aggiungere nuove funzionalità modificando la matrice del Nutchip.

Che ne dite di...

-) aggiungere dei pulsanti per scegliere tempi diversi o resettare il tempo
-) aggiungere un modulo ricevitore e sostituire il pulsante SW1 con un telecomando (potete fare le prove senza dovere acquistare il modulo: basta usare il telecomando virtuale)
-) utilizzare le altre uscite libere per accendere dei LED: verde = OK, giallo = tempo quasi scaduto, rosso = tempo scaduto



[download telecomando virtuale](#)

Una fotocellula intelligente

[Un occhio elettronico](#) | [Schema base \(1\)](#) | [Aggiungere un'isteresi](#) | [Montaggio sperimentale](#)
[Evoluzione di un circuito \(2\)](#) | [Sfruttiamo il cervello](#)

Un occhio elettronico

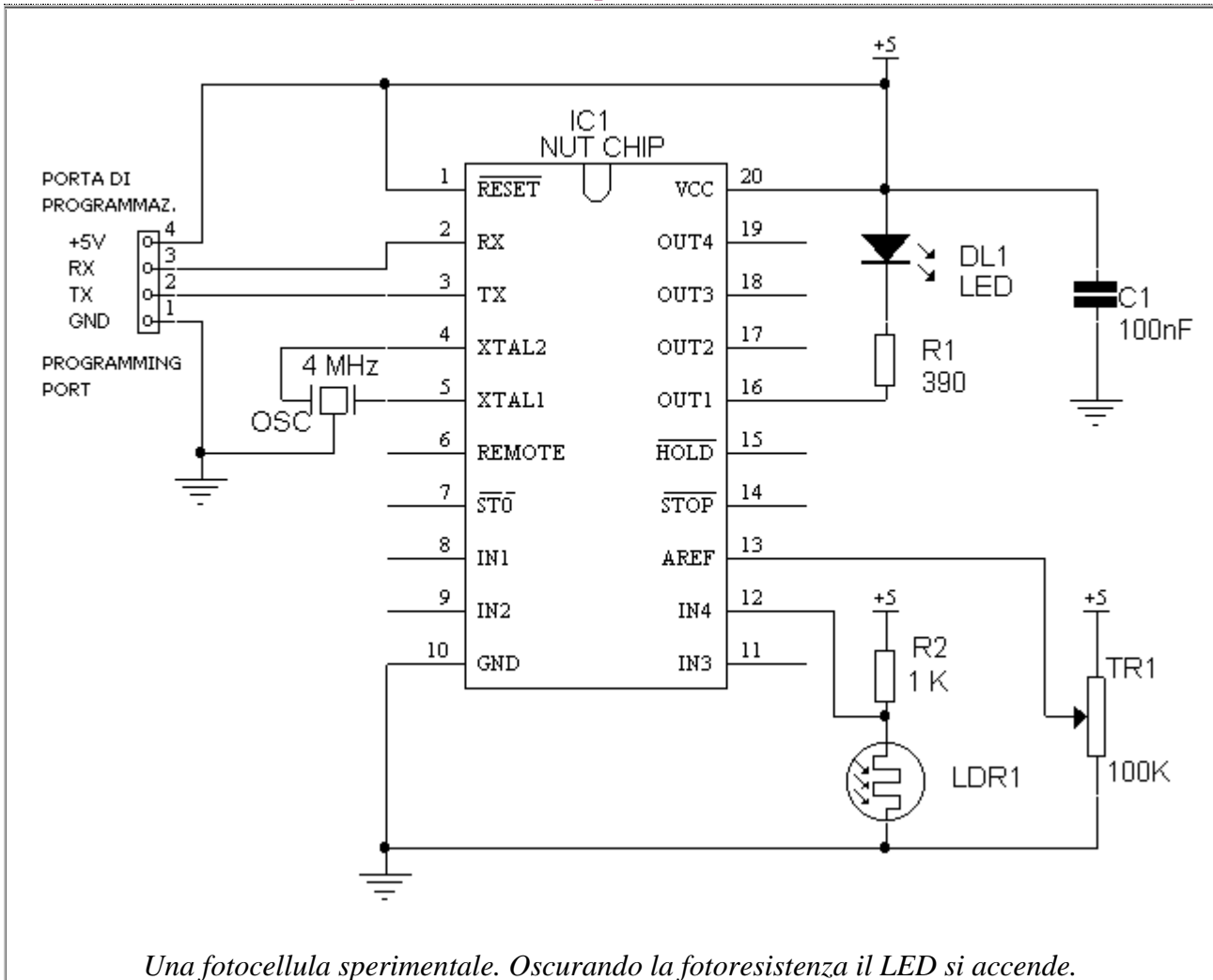
La fotocellula è un circuito utile e dalle infinite applicazioni. Il suo occhio elettronico è capace di accendere le luci all'imbrunire o entrando in auto una galleria o a piedi in un lungo corridoio. E' in grado anche di rivelare il passaggio di un veicolo, di sentire la presenza di una fiamma, e persino di riconoscere la mancanza di un'etichetta su un prodotto. Il Nutchip si interfaccia direttamente ad un sensore di luce (LDR) che è il cuore della fotocellula; il suo cervello logico realizza poi mille automatismi senza che sia necessario aggiungere nulla di esterno.

Ecco quindi un simulatore di presenza che si accende automaticamente la sera per un'ora in soggiorno, poi un'altra ora in cucina, ed infine accende le luci in camera da letto per poi spegnerle prima che faccia troppo tardi. Un compito che il Nutchip svolge egregiamente!

1) Schema base

Lo schema base della fotocellula utilizza l'ingresso analogico (comparatore) del Nutchip. Questo ingresso è gestito in comune con l'ingresso digitale IN4, e dovremo ricordare di attivarlo al momento della programmazione del chip.

Il comparatore funziona confrontando la tensione all'ingresso IN4 con quella di riferimento presente all'ingresso AREF; se la tensione applicata su IN4 è maggiore di quella applicata su AREF, allora IN4 viene visto a livello logico uno; altrimenti a livello logico zero. Per esempio, immaginiamo di regolare il trimmer TR1 per dare 2 volt sul piedino AREF. Immaginiamo anche che in condizioni di luce la resistenza dell'LDR sia di 1 kohm. Allora al piedino IN4 arriveranno 2,5 volt, per cui a livello logico sarà IN4 = livello alto, cioè uno. Oscurando la fotocellula la resistenza salirà a 10 kohm; la tensione sul piedino IN4 crolla ben al di sotto dei 2 volt presenti sul piedino AREF. In queste condizioni, IN4 sarà visto a livello basso, cioè zero.



Nella pratica regoleremo il trimmer TR1 al livello di sensibilità ottimale per l'applicazione che vogliamo realizzare, cioè un pò prima che la fotocellula scatti. In tal modo, anche una piccola variazione di luce farà scattare l'uscita.

Aggiungere un'isteresi

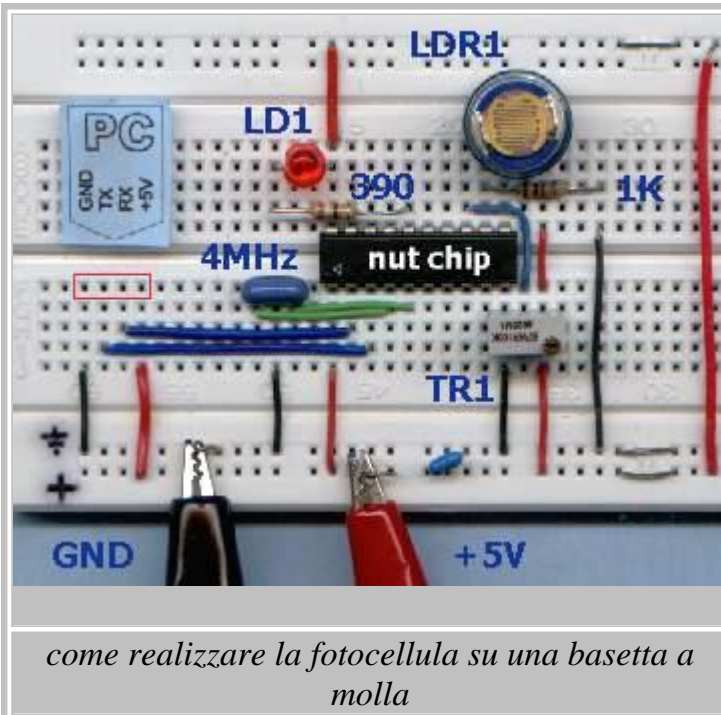
A volte non è desiderabile avere un funzionamento troppo sensibile. Ad esempio, in una giornata parzialmente nuvolosa non è gradevole che le luci si accendano e si spengano in corrispondenza di ogni piccola nuvoletta! Per ovviare a questo inconveniente, tipico di tutti gli automatismi, si introduce solitamente un'isteresi.

Ad esempio, si potrebbe fare in modo che appena la fotocellula scatta, la sua sensibilità venga ridotta. In tal modo, per fare rilasciare la fotocellula sarebbe necessario un considerevole aumento della luminosità, ben al di sopra rispetto al valore che l'ha fatta intervenire. Un tale tipo di isteresi si chiama isteresi sul livello, perchè avviene variando il livello di luminosità che provoca l'attacco o il distacco del meccanismo.

Con i Nutchips è conveniente adottare un'isteresi sul tempo, che non richiede risorse hardware ma soltanto di inserire un timeout nella tabella della verità. Infatti, se il nostro scopo è evitare un'eccessivo attacca/stacca della fotocellula quando la luminosità oscilla attorno al valore di intervento, imponendo che la fotocellula resti accesa per un intervallo fisso centeremo il risultato. Al termine del timeout torneremo l'uscita tornerà ad essere controllata dall'ingresso analogico.

Montaggio sperimentale

Per realizzare la fotocellula bastano pochissimi componenti, perciò è facile assemblarla su una basetta sperimentale a molla.



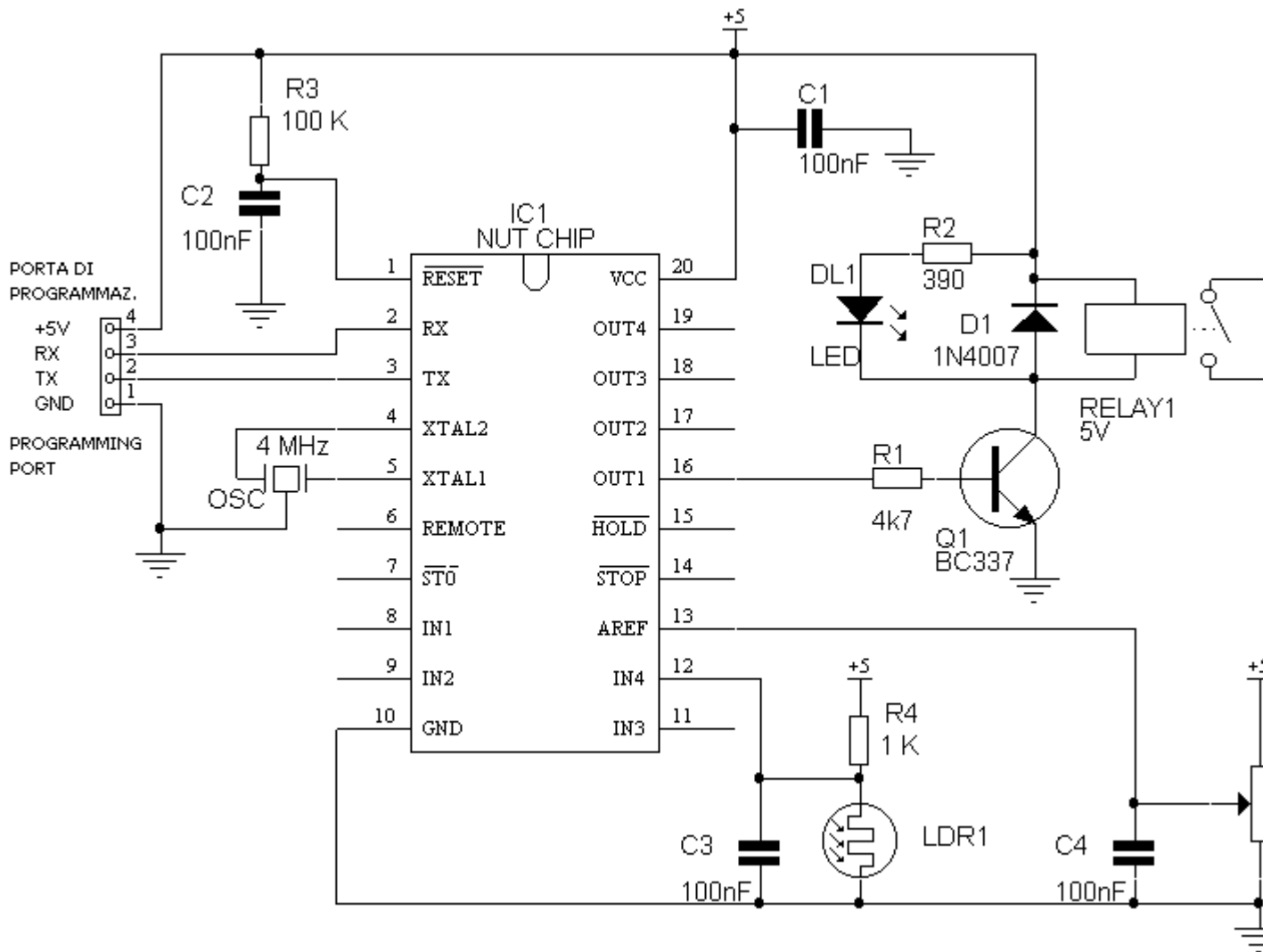
Elenco componenti:

- J 1 LED rosso
- J 1 resistenza 390 ohm
- J 1 resistenza 1000 ohm
- J trimmer 100 kilohm (pref. multigiri)
- J 1 cond. 100 nF
- J 1 fotoresistenza (LDR)
- J 1 ris. ceramico 3 piedini da 4 MHz
- J 1 Nutchip

2) Evoluzione di un circuito

E' il momento di preparare il circuito ad uscire dal laboratorio. La fotocellula è un apparecchio destinato a funzionare ininterrottamente per mesi: dobbiamo pensare prima di tutto all'affidabilità. In un periodo di tempo molto lungo, è inevitabile avere fluttuazioni di corrente ed anche blackout. Aggiungiamo quindi un circuito di reset composto da una rete resistenza-condensatore, R3 e C2. Anche gli ingressi del comparatore vanno ripuliti, per evitare dei falsi azionamenti causati dagli onnipresenti disturbi. I condensatori C3 e C4 servono proprio ad appianare i picchi e a livellare le tensione agli ingressi del comparatore. Affianchiamo poi al LED un relè, per pilotare il nostro carico. Ricordiamo di dimensionare l'ampereaggio del relè in base al carico da pilotare, lasciando spazio per un'abbondante "scorta" di corrente. Ad esempio, una lampada da 100 W al momento dell'accensione arriva ad assorbire fino a 1000 W! Una regola di buon senso consiste nell'acquistare un relè la cui corrente sia almeno doppia rispetto a quella effettivamente necessaria. I relè da 5 volt (come quelli utilizzati qui) sono comodi, ma talvolta sono difficili da reperire per le correnti maggiori: potete sempre modificare lo stadio di uscita per pilotare un relè a 12 volt, come indicato nella raccolta di circuiti base.

Queste protezioni basteranno per un utilizzo "casalingo" della nostra fotocellula.



La fotocellula esce dal laboratorio:

-) Il LED è affiancato da un relè (il funzionamento è inverso rispetto allo schema precedente)
-) Il reset ha una propria rete di ritardo costituita da una resistenza ed un condensatore, per maggior affidabilità
-) Le tensioni analogiche da misurare sono "ripulite" da C3 e C4.

Sfruttiamo il cervello

La caratteristica più sfiziosa di questa fotocellula è la sua programmabilità, non ottenibile con le fotocellule tradizionali.

Eccone alcune interessanti applicazioni:

-) **Accensione a ricarica** - Una scala mobile è azionata dall'avvicinarsi delle persone. Ogni volta che la fotocellula sente la presenza di una persona, si attiva per un tempo sufficiente per la salita o la discesa. Il timer si "ricarica" finchè c'è gente (questa funzione si chiama tecnicamente "allungamento di impulso"). Un'accensione di questo tipo è adatta anche per le luci di un lungo corridoio o di una

toilette. In quest'ultimo caso è utile collegare un secondo relè che accenda un ventilatore per un tempo diverso.

- J) **Contapezzi** - Alla fine di un nastro trasportatore vogliamo un occhio controllare automaticamente che tutte le confezioni contengano esattamente tre bottiglie di vino. Se manca una bottiglia, il relè attiva un segnale d'allarme per avvisare di scartare la confezione. Il compito è reso facile dal colore scuro del vetro che contrasta contro il fondo chiaro della confezione: la fotocellula dovrà contare tre impulsi vicini fra loro (le tre bottiglie), poi una pausa lunga (lo spazio fra una confezione e l'altra sul nastro trasportatore). Se gli impulsi sono solo due, o uno, o se la pausa si allunga troppo, si fa scattare il relè di allarme.
- J) **Annaffiatore autoregolante** - Perché montare un display con mille pulsanti quando il momento giusto è semplicemente "*quattro ore dopo il tramonto*"? C'è un metodo più furbo per comandare l'irrigazione del vostro giardino che non realizzare un sofisticato orologio solo per azionare un relè!

Una fotocellula intelligente

[Un occhio elettronico](#) | [Schema base \(1\)](#) | [Aggiungere un'isteresi](#) | [Montaggio sperimentale](#)
[Evoluzione di un circuito \(2\)](#) | [Sfruttiamo il cervello](#)

Un occhio elettronico

La fotocellula è un circuito utile e dalle infinite applicazioni. Il suo occhio elettronico è capace di accendere le luci all'imbrunire o entrando in auto una galleria o a piedi in un lungo corridoio. E' in grado anche di rivelare il passaggio di un veicolo, di sentire la presenza di una fiamma, e persino di riconoscere la mancanza di un'etichetta su un prodotto. Il Nutchip si interfaccia direttamente ad un sensore di luce (LDR) che è il cuore della fotocellula; il suo cervello logico realizza poi mille automatismi senza che sia necessario aggiungere nulla di esterno.

Ecco quindi un simulatore di presenza che si accende automaticamente la sera per un'ora in soggiorno, poi un'altra ora in cucina, ed infine accende le luci in camera da letto per poi spegnerle prima che faccia troppo tardi. Un compito che il Nutchip svolge egregiamente!

1) Schema base

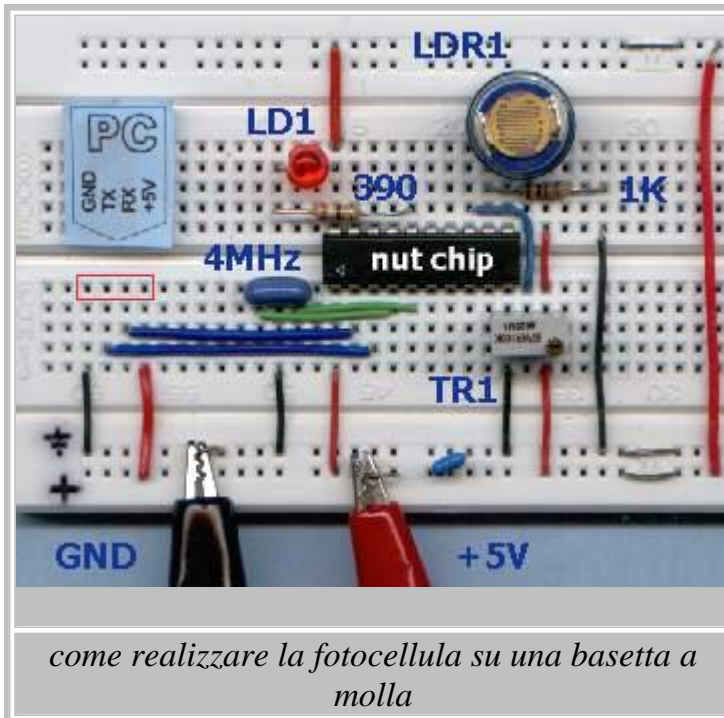
Lo schema base della fotocellula utilizza l'ingresso analogico (comparatore) del Nutchip. Questo ingresso è gestito in comune con l'ingresso digitale IN4, e dovremo ricordare di attivarlo al momento della programmazione del chip.

Il comparatore funziona confrontando la tensione all'ingresso IN4 con quella di riferimento presente all'ingresso AREF; se la tensione applicata su IN4 è maggiore di quella applicata su AREF, allora IN4 viene visto a livello logico uno; altrimenti a livello logico zero. Per esempio, immaginiamo di regolare il trimmer TR1 per dare 2 volt sul piedino AREF. Immaginiamo anche che in condizioni di luce la resistenza dell'LDR sia di 1 kohm. Allora al piedino IN4 arriveranno 2,5 volt, per cui a livello logico sarà IN4 = livello alto, cioè uno. Oscurando la fotocellula la resistenza salirà a 10

intervento, imponendo che la fotocellula resti accesa per un intervallo fisso centeremo il risultato. Al termine del timeout torneremo l'uscita tornerà ad essere controllata dall'ingresso analogico.

Montaggio sperimentale

Per realizzare la fotocellula bastano pochissimi componenti, perciò è facile assemblarla su una basetta sperimentale a molla.



Elenco componenti:

- J 1 LED rosso
- J 1 resistenza 390 ohm
- J 1 resistenza 1000 ohm
- J trimmer 100 kilohm (pref. multigiri)
- J 1 cond. 100 nF
- J 1 fotoresistenza (LDR)
- J 1 ris. ceramico 3 piedini da 4 MHz
- J 1 Nutchip

2) Evoluzione di un circuito

E' il momento di preparare il circuito ad uscire dal laboratorio. La fotocellula è un apparecchio destinato a funzionare ininterrottamente per mesi: dobbiamo pensare prima di tutto all'affidabilità. In un periodo di tempo molto lungo, è inevitabile avere fluttuazioni di corrente ed anche blackout. Aggiungiamo quindi un circuito di reset composto da una rete resistenza-condensatore, R3 e C2. Anche gli ingressi del comparatore vanno ripuliti, per evitare dei falsi azionamenti causati dagli onnipresenti disturbi. I condensatori C3 e C4 servono proprio ad appianare i picchi e a livellare le tensione agli ingressi del comparatore. Affianchiamo poi al LED un relè, per pilotare il nostro carico. Ricordiamo di dimensionare l'ampereaggio del relè in base al carico da pilotare, lasciando spazio per un'abbondante "scorta" di corrente. Ad esempio, una lampada da 100 W al momento dell'accensione arriva ad assorbire fino a 1000 W! Una regola di buon senso consiste nell'acquistare un relè la cui corrente sia almeno doppia rispetto a quella effettivamente necessaria. I relè da 5 volt (come quelli utilizzati qui) sono comodi, ma talvolta sono difficili da reperire per le correnti maggiori: potete sempre modificare lo stadio di uscita per pilotare un relè a 12 volt, come indicato nella raccolta di circuiti base.

Queste protezioni basteranno per un utilizzo "casalingo" della nostra fotocellula.

toilette. In quest'ultimo caso è utile collegare un secondo relè che accenda un ventilatore per un tempo diverso.

- J) **Contapezzi** - Alla fine di un nastro trasportatore vogliamo un occhio controllare automaticamente che tutte le confezioni contengano esattamente tre bottiglie di vino. Se manca una bottiglia, il relè attiva un segnale d'allarme per avvisare di scartare la confezione. Il compito è reso facile dal colore scuro del vetro che contrasta contro il fondo chiaro della confezione: la fotocellula dovrà contare tre impulsi vicini fra loro (le tre bottiglie), poi una pausa lunga (lo spazio fra una confezione e l'altra sul nastro trasportatore). Se gli impulsi sono solo due, o uno, o se la pausa si allunga troppo, si fa scattare il relè di allarme.

- J) **Anaffiatore autoregolante** - Perché montare un display con mille pulsanti quando il momento giusto è semplicemente "*quattro ore dopo il tramonto*"? C'è un metodo più furbo per comandare l'irrigazione del vostro giardino che non realizzare un sofisticato orologio solo per azionare un relè!

Radiocomando a 3 canali

convertibile anche a raggi infrarossi e comandabile da PC!!

[Per tutte le stagioni](#) | [Schema elettrico](#) | [Tavola della verità](#) | [Collegamento](#) | [Trasformazione ad infrarossi](#) | [Controllo tramite PC](#) | [Bravo Pasquale!](#)

Per tutte le stagioni

Il telecomando è uno degli accessori più utili e versatili. Semplice e familiare, è in grado di rendere la nostra vita più confortevole.

Come ci ha confidato un amico elettricista, oggi è conveniente e rapido installare un *telecomando* dove fino a ieri sarebbe stato impossibile o molto laborioso installare nuovi fili.

E' un progetto *evergreen*, un sempreverde adatto in tutte le stagioni:

ESTATE: Che caldo, è estate, che bello accendere il ventilatore direttamente dal letto!

INVERNO: Che freddo, che bello accendere l'albero di Natale in giardino senza uscire di casa!

INVERNO: Notte e giorno sul presepe, con tre canali l'immaginazione vola!

ESTATE: L'irrigazione del prato con si accende come per magia!

TUTTO L'ANNO: il telecomando accende e spegne caldaie, luci, elettrodomestici....

Con i suoi tre canali, questo telecomando va oltre i normali e semplici telecomandi. Infatti nel suo cuore batte un Nutchip, che lo rende facilmente **modificabile** secondo le nostre esigenze.

Schema elettrico

Lo schema elettrico è molto semplice e comprende, oltre al Nutchip, tre relè con relativi transistor di pilotaggio e LED di segnalazione, il ricevitore radio. I dati escono dal piedino OUT del ricevitore ed entrano nel piedino REMOTE del telecomando. Il piedino di -RESET è collegato ad una rete di ritardo RC esterna. Se dovete montare il telecomando dove sia richiesto un servizio ininterrotto 24 ore su 24 oppure in ambienti elettricamente inquinati allora sarà bene utilizzare un integrato apposito per il RESET. Notate come i piedini di ingresso non utilizzati siano collegati al positivo: è una buona abitudine che rende più difficile la vita ai disturbi. Anche i condensatori C2 e C3 servono per "ripulire" la tensione di alimentazione da eventuali disturbi radio che dovessero infiltrarsi verso il ricevitore: dovranno essere montati il più vicino possibile ai terminali del ricevitore, con fili brevissimi verso la massa.

Tutto il circuito viene alimentato a 5 volt, compresi i relè. Siccome i relè da 12 volt continui sono più facilmente reperibili potreste desiderare di montare questi ultimi al posto di quelli a 5 volt: in tal caso no problem, basterà distaccare il terminale dei relè che va alla +5volt per collegarli ad un alimentatore che fornisca anche la 12 volt.

La tavola della verità

Occorre innanzitutto stabilire quali tasti servano per accendere le uscite e quali per spegnerle. Decidiamo che ai tasti siano assegnate le funzioni che seguono:

- | | |
|----------------------------|---------------------------|
|) key1: accende l'uscita 1 |) key4: spegne l'uscita 1 |
|) key2: accende l'uscita 2 |) key5: spegne l'uscita 2 |
|) key3: accende l'uscita 3 |) key6: spegne l'uscita 3 |

La tabella degli stati di questo radiocomando ha otto stati: in pratica ad ogni combinazione delle uscite corrisponde uno stato. Per ogni stato ci sono solo tre possibili condizioni: immaginiamo di trovarci ad **esempio** nello stato st03. Leggendo le uscite corrispondenti a questo stato sulla tavola, scopriamo che la prima e la seconda uscita saranno accese, mentre il terzo relè sarà diseccitato. In queste condizioni possiamo:

-) premere key4 per *spegner*e la prima uscita
-) premere key5 per *spegner*e la seconda uscita
-) premere key3 per *accender*e la terza uscita.

Key1, key2 e key6 non ci interessano perchè la prima e la seconda uscita sono già accese mentre la terza è già spenta; la pressione di tali tasti sarebbe senza alcun effetto comunque, perchè accendere una luce già accesa o spegnerne una già spenta non ha alcun senso. Immaginiamo invece di premere key5; allora vogliamo spegnere il secondo relè, cosa che facciamo portandoci nello stato st01. Inoltre lo stato st01 ignora le successive ed inevitabili ripetizioni del tasto key5, che infatti non compare in st01.

Collegamento

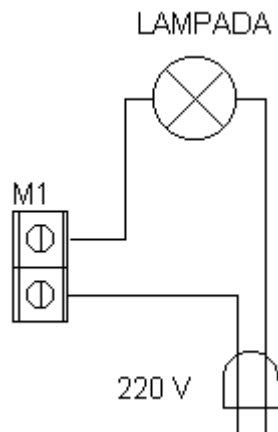
I contatti dei relè sono portati all'esterno tramite i tre

state	output 1...4	input 1...4	remote	next
st00	0 0 0 0	- - - -	key 1 key 2 key 3	st01 st02 st04
st01	1 0 0 0	- - - -	key 4 key 2 key 3	st00 st03 st05
st02	0 1 0 0	- - - -	key 1 key 5 key 3	st03 st00 st06
st03	1 1 0 0	- - - -	key 4 key 5 key 3	st02 st01 st07
st04	0 0 1 0	- - - -	key 1 key 2 key 6	st05 st06 st00
st05	1 0 1 0	- - - -	key 4 key 2 key 6	st04 st07 st01
st06	0 1 1 0	- - - -	key 1 key 5 key 6	st07 st04 st02
st07	1 1 1 0	- - - -	key 4 key 5 key 6	st06 st05 st03

tavola della verità del comando a distanza, contenuta nel file remote.nut. Non dimenticate di impostare i codici relativi ai tasti del telecomando in vostro possesso.

Bravo Pasquale!

morsetti M1, M2, M3. Qui potremo collegare lampade, radio, piccoli elettrodomestici, come se si trattasse di un comune interruttore. Controllate sempre però che la corrente assorbita dal carico sia inferiore a quella massima ammessa dal relè, e mantenete un ampio margine di sicurezza. Utilizzate ad esempio un relè la cui corrente massima sia almeno il doppio di quella effettivamente assorbita: questo eviterà surriscaldamenti del relè ed il pericolo di fusioni, archi ed incendi.



Se non trovate un relè a 12 volt in grado di commutare la corrente richiesta, potete usare l'uscita per pilotare un relè di potenza esterno a 220 volt.

Importante: Per la vostra incolumità, collegate il carico solo ed esclusivamente con l'alimentazione spenta! Ricollegate l'alimentazione solo quando i collegamenti siano stati completati e ricontrollati, e solo dopo che il circuito sia stato collocato in una scatola di materiale isolante per evitare qualsiasi contatto accidentale con la corrente elettrica.

Trasformazione ad infrarossi

La trasformazione del comando a distanza da radiofrequenza a raggi infrarossi richiede soltanto di sostituire il ricevitore radio con un modulo ricevitore infrarossi (per il collegamento vedi la [lampada per tv](#)). Ovviamente dovremo disporre anche del telecomando ad infrarossi, reperibile un pò ovunque. Dovremo quindi riprogrammare il Nutchip dopo avere selezionato il telecomando adeguato dalla pagina di programmazione. La tavola ed il resto dello restano identici, a voi la scelta della "portante" che più vi piace!

*Un'applicazione interessante e fantasiosa è quella realizzata dall'amico Pasquale di Napoli, la città dove il caffè è un'arte. Ha comandato la macchina del caffè, in modo da poterla accendere al mattino **direttamente dal letto**. Il tempo per un'altro pisolino, ed appena si scende dal letto tutto è pronto per un fumante caffè! Siccome non disponeva di un relè in grado di sopportare la corrente richiesta, ha collegato all'uscita un secondo relè di elevata potenza a 220 volt, in grado di sopportare il carico richiesto. Ha poi **modificato** la tabella della verità in modo che la macchina si spenga automaticamente al massimo dopo venti minuti, in caso ci si scordasse di spegnerla. . Ha anche **collegato** un pulsante all'ingresso -ST0 per spegnerla manualmente direttamente dalla centralina. Bravo Pasquale, a quando un invito per un caffè a casa tua?*

Controllo con PC via seriale

Questo schema si presta anche per controllare i relè attraverso il PC. Non serve **nessuna modifica**: basta lasciare collegata [l'interfaccia seriale](#) al PC ed utilizzare il software Nutchip Commander scaricabile gratuitamente dalla [pagina dei download](#).

Il telecomando "virtuale" (sullo schermo del PC) e quello "reale" funzionano assieme, senza interferirsi fra loro.

Se vi basta il solo controllo da PC allora non perdetevi il progetto della [scheda PC a 3 canali](#): c'è anche il circuito stampato!



Radiocomando a 3 canali

convertibile anche a raggi infrarossi e comandabile da PC!!

[Per tutte le stagioni](#) | [Schema elettrico](#) | [Tavola della verità](#) | [Collegamento](#) | [Trasformazione ad infrarossi](#) | [Controllo tramite PC](#) | [Bravo Pasquale!](#)

Per tutte le stagioni

Il telecomando è uno degli accessori più utili e versatili. Semplice e familiare, è in grado di rendere la nostra vita più confortevole.

Come ci ha confidato un amico elettricista, oggi è conveniente e rapido installare un *telecomando* dove fino a ieri sarebbe stato impossibile o molto laborioso installare nuovi fili.

E' un progetto *evergreen*, un sempreverde adatto in tutte le stagioni:

ESTATE: Che caldo, è estate, che bello accendere il ventilatore direttamente dal letto!

INVERNO: Che freddo, che bello accendere l'albero di Natale in giardino senza uscire di casa!

INVERNO: Notte e giorno sul presepe, con tre canali l'immaginazione vola!

ESTATE: L'irrigazione del prato con si accende come per magia!

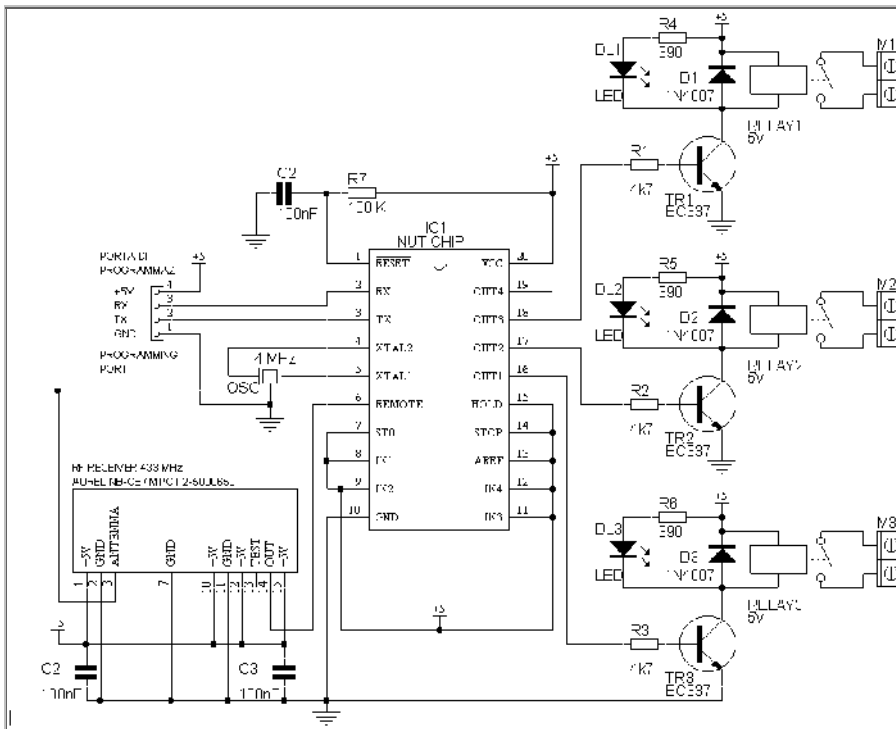
TUTTO L'ANNO: il telecomando accende e spegne caldaie, luci, elettrodomestici....

Con i suoi tre canali, questo telecomando va oltre i normali e semplici telecomandi. Infatti nel suo cuore batte un Nutchip , che lo rende facilmente **modificabile** secondo le nostre esigenze.

Schema elettrico

Lo schema elettrico è molto semplice e comprende, oltre al Nutchip, tre relè con relativi transistor di pilotaggio e LED di segnalazione, il ricevitore radio. I dati escono dal piedino OUT del ricevitore ed entrano nel piedino REMOTE del telecomando. Il piedino di -RESET è collegato ad una rete di ritardo RC esterna. Se dovete montare il telecomando dove sia richiesto un servizio ininterrotto 24 ore su 24 oppure in ambienti elettricamente inquinati allora sarà bene utilizzare un integrato apposito per il RESET. Notate come i piedini di ingresso non utilizzati siano collegati al positivo: è una buona abitudine che rende più difficile la vita ai disturbi. Anche i condensatori C2 e C3 servono per "ripulire" la tensione di alimentazione da eventuali disturbi radio che dovessero infilarsi verso il ricevitore: dovranno essere montati il più vicino possibile ai terminali del ricevitore, con fili brevissimi verso la massa.

Tutto il circuito viene alimentato a 5 volt, compresi i relè. Siccome i relè da 12 volt continui sono più facilmente reperibili potreste desiderare di montare questi ultimi al posto di quelli a 5 volt: in tal caso no problem, basterà distaccare il terminale dei relè che va alla +5volt per collegarli ad un alimentatore che fornisca anche la 12 volt.



La tavola della verità

Occorre innanzitutto stabilire quali tasti servano per accendere le uscite e quali per spegnerle. Decidiamo che ai tasti siano assegnate le funzioni che seguono:

- | | |
|----------------------------|---------------------------|
| J key1: accende l'uscita 1 | J key4: spegne l'uscita 1 |
| J key2: accende l'uscita 2 | J key5: spegne l'uscita 2 |
| J key3: accende l'uscita 3 | J key6: spegne l'uscita 3 |

La tabella degli stati di questo radiocomando ha otto stati: in pratica ad ogni combinazione delle uscite corrisponde uno stato. Per ogni stato ci sono solo tre possibili condizioni: immaginiamo di trovarci ad **esempio** nello stato st03. Leggendo le uscite corrispondenti a questo stato sulla tavola, scopriamo che la prima e la seconda uscita saranno accese, mentre il terzo relè sarà diseccitato. In queste condizioni possiamo:

- J premere key4 per *spegnere* la prima uscita
- J premere key5 per *spegnere* la seconda uscita
- J premere key3 per *accendere* la terza uscita.

Key1, key2 e key6 non ci interessano perchè la prima e la seconda uscita sono già accese mentre la terza è già spenta; la pressione di tali tasti sarebbe senza alcun effetto comunque, perchè accendere una luce già accesa o spegnerne una già spenta non ha alcun senso. Immaginiamo invece di premere key5; allora vogliamo spegnere il secondo relè, cosa che facciamo portandoci nello stato st01. Inoltre lo stato st01 ignora le successive ed inevitabili ripetizioni del tasto key5, che infatti non compare in st01.

Collegamento

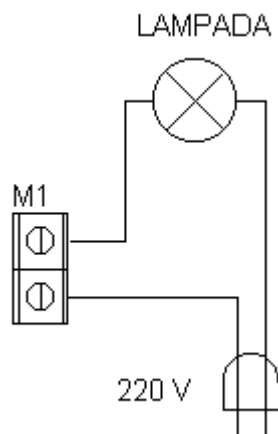
I contatti dei relè sono portati all'esterno tramite i tre

state	output 1...4	input 1...4	remote	next
st00	0 0 0 0	- - - -	key 1 key 2 key 3	st01 st02 st04
st01	1 0 0 0	- - - -	key 4 key 2 key 3	st00 st03 st05
st02	0 1 0 0	- - - -	key 1 key 5 key 3	st03 st00 st06
st03	1 1 0 0	- - - -	key 4 key 5 key 3	st02 st01 st07
st04	0 0 1 0	- - - -	key 1 key 2 key 6	st05 st06 st00
st05	1 0 1 0	- - - -	key 4 key 2 key 6	st04 st07 st01
st06	0 1 1 0	- - - -	key 1 key 5 key 6	st07 st04 st02
st07	1 1 1 0	- - - -	key 4 key 5 key 6	st06 st05 st03

tavola della verità del comando a distanza, contenuta nel file remote.nut. Non dimenticate di impostare i codici relativi ai tasti del telecomando in vostro possesso.

Un'applicazione interessante e fantasiosa è quella realizzata

morsetti M1, M2, M3. Qui potremo collegare lampade, radio, piccoli elettrodomestici, come se si trattasse di un comune interruttore. Controllate sempre però che la corrente assorbita dal carico sia inferiore a quella massima ammessa dal relè, e mantenete un ampio margine di sicurezza. Utilizzate ad esempio un relè la cui corrente massima sia almeno il doppio di quella effettivamente assorbita: questo eviterà surriscaldamenti del relè ed il pericolo di fusioni, archi ed incendi.



Se non trovate un relè a 12 volt in grado di commutare la corrente richiesta, potete usare l'uscita per pilotare un relè di potenza esterno a 220 volt.

Importante: Per la vostra incolumità, collegate il carico solo ed esclusivamente con l'alimentazione spenta! Ricollegate l'alimentazione solo quando i collegamenti siano stati completati e ricontrollati, e solo dopo che il circuito sia stato collocato in una scatola di materiale isolante per evitare qualsiasi contatto accidentale con la corrente elettrica.

Trasformazione ad infrarossi

La trasformazione del comando a distanza da radiofrequenza a raggi infrarossi richiede soltanto di sostituire il ricevitore radio con un modulo ricevitore infrarossi (per il collegamento vedi la [lampada per tv](#)). Ovviamente dovremo disporre anche del telecomando ad infrarossi, reperibile un pò ovunque. Dovremo quindi riprogrammare il Nutchip dopo avere selezionato il telecomando adeguato dalla pagina di programmazione. La tavola ed il resto dello restano identici, a voi la scelta della "portante" che più vi piace!

*dall'amico Pasquale di Napoli, la città dove il caffè è un'arte. Ha comandato la macchina del caffè, in modo da poterla accendere al mattino **direttamente dal letto**. Il tempo per un'altro pisolino, ed appena si scende dal letto tutto è pronto per un fumante caffè! Siccome non disponeva di un relè in grado di sopportare la corrente richiesta, ha collegato all'uscita un secondo relè di elevata potenza a 220 volt, in grado di sopportare il carico richiesto. Ha poi **modificato** la tabella della verità in modo che la macchina si spenga automaticamente al massimo dopo venti minuti, in caso ci si scordasse di spegnerla. . Ha anche **collegato** un pulsante all'ingresso -ST0 per spegnerla manualmente direttamente dalla centralina. Bravo Pasquale, a quando un invito per un caffè a casa tua?*

Controllo con PC via seriale

Questo schema si presta anche per controllare i relè attraverso il PC. Non serve **nessuna modifica**: basta lasciare collegata [l'interfaccia seriale](#) al PC ed utilizzare il software Nutchip Commander scaricabile gratuitamente dalla [pagina dei download](#).

Il telecomando "virtuale" (sullo schermo del PC) e quello "reale" funzionano assieme, senza interferirsi fra loro.

Se vi basta il solo controllo da PC allora non perdetevi il progetto della [scheda PC a 3 canali](#): c'è anche il circuito stampato!



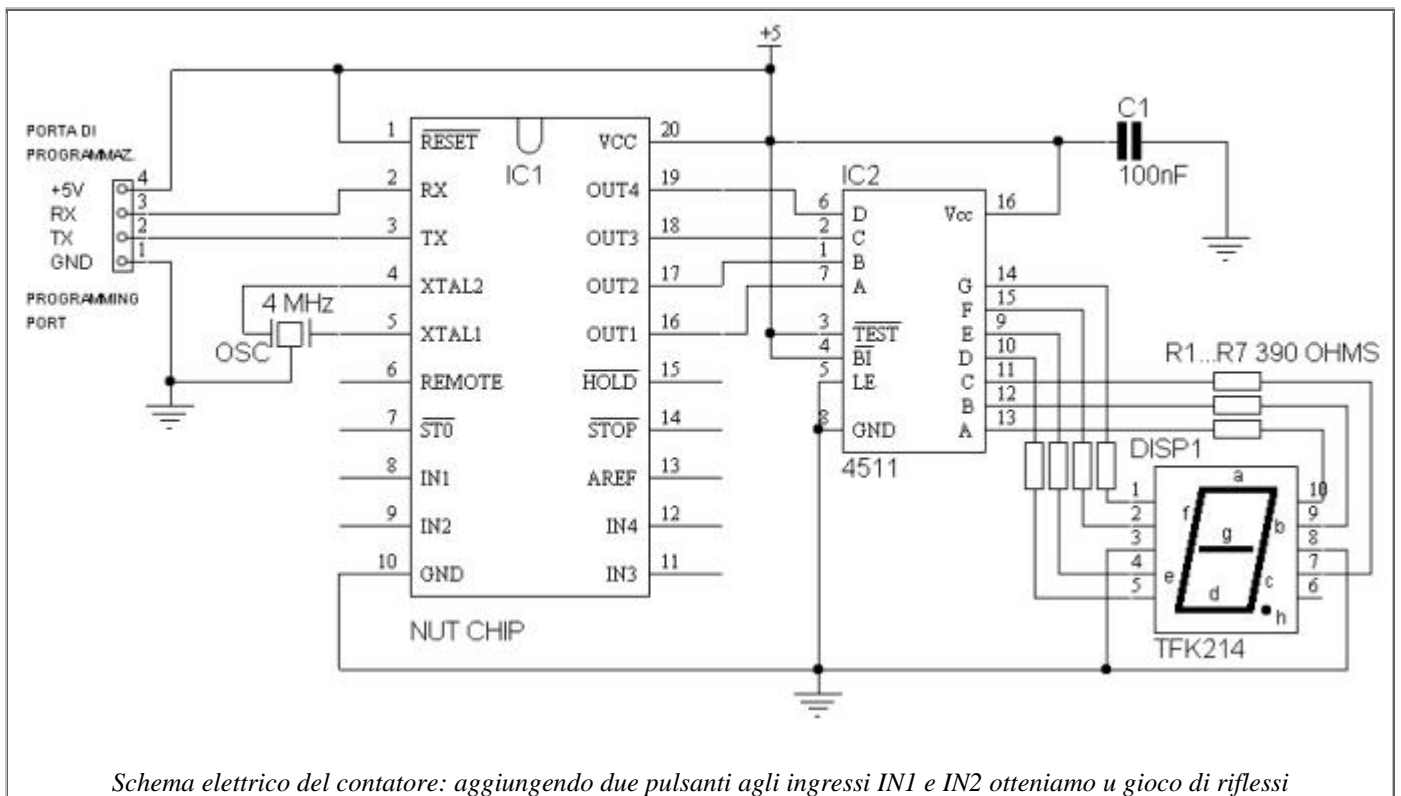
Il telecomando "virtuale" sullo schermo del PC. I LED rossi visualizzano le uscite.

Contatore

[Il display a 7 segmenti](#) | [Il pilota display 4511](#) | [Realizzazione](#) | [Tavola della verità](#) | [Come un dado](#)

Questo progetto introduce una funzione fondamentale dell'elettronica digitale: il *conteggio*.

Realizzando un semplice contatore che avanza automaticamente a passi di un secondo impareremo come sono fatti e come funzionano i display a sette segmenti come vanno pilotati. Chi sapesse già tutto dell'argomento, può saltare direttamente alla seconda parte di questo progetto, in cui si sfrutta il contatore per realizzare un [gioco di riflessi](#).



Il display a 7 segmenti.

Un display a sette segmenti è costruito riunendo in un unico contenitore 7 led. Ogni led è sagomato in modo da illuminare un "segmento": illuminando due o più segmenti assieme potremo leggere sul

display tutte le cifre da 0 a 9.

I segmenti sono indicati convenzionalmente con le lettere da A fino a G: potete vedere la disposizione dei segmenti sul display dallo schema elettrico.

Per esempio, il numero due si ottiene illuminando i segmenti A, B, G, E, D, mentre un tre si ottiene illuminando i segmenti A, B, C, D, G; per avere un otto invece accenderemotutti i segmenti da A a G.

A dispetto del nome, tutti i display oggi hanno un ottavo led che illumina il puntino decimale: questo segmento è indicato di solito dalla lettera H o dalla sigle DP (decimal point).

Per ridurre il numero dei piedini, all'interno del display i catodi di tutti i led sono collegati assieme e portati all'esterno su di un unico pin; sugli altri pin troveremo invece tutti gli anodi dei led da A fino ad H.. Questi display si chiamano "a catodo comune". Il display TFK214 utilizzato nel nostro progetto è proprio di questo tipo. Potete sostituirlo con qualsiasi altro display a sette segmenti purchè sia sempre a catodo comune: il tal caso però dovrete rifare quasi sicuramente i collegamenti perchè la piedinatura di solito non corrisponde da un tipo di display ad un altro. Ecco come è collegato il TFK214:

piedino	segmento
1	G
2	F
3	catodo comune
4	E
5	D
6	H (DP)
7	C
8	catodo comune
9	B
10	A

pinout del display TFK214

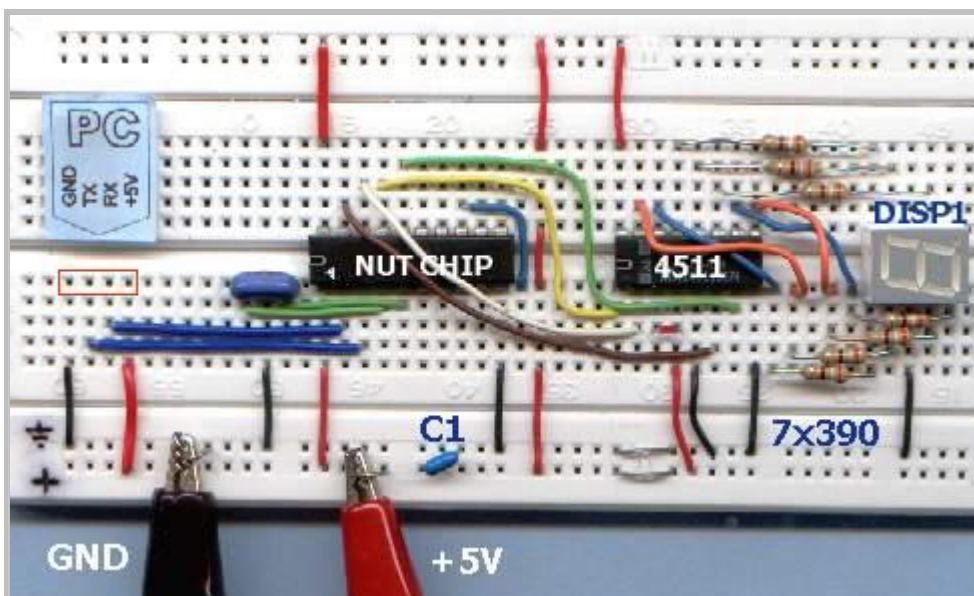
Il pilota display 4511

Il circuito integrato CD4511 è molto diffuso ed economico. La sua funzione è quella di "leggere" un numero binari sui suoi 4 ingressi A-B-C-D e di eccitare le uscite A-B-C-D-E-F-G in modo da accendere un display a catodo comune formando la cifra corrispondente. Come il NUT, anche il CD4511 ha al suo interno una tavola della verità, eccola:

numero	binario	segmenti	numero	binario	segmenti
0	0000	A,B,C,D,E,F	8	1000	A,B,C,D,E,F,G
1	0001	B,C	9	1001	A,B,C,F,G
2	0010	A,B,G,E,D	10	1010	A,B,C,D,E,F
3	0011	A,B,C,D,G	11	1011	spento
4	0100	F,G,B,C	12	1100	spento
5	0101	A,F,G,C,D	13	1101	spento
6	0110	F,G,C,D,E	14	1110	spento
7	0111	A,B,C	15	1111	spento

Realizzazione

Il montaggio non dovrebbe presentare alcuna difficoltà. E' però necessaria molta cura per non invertire i componenti polarizzati (display, NUTCHIP e 4511). Se avete un altro display a catodo comune potete utilizzarlo al posto del TFK514; in tal caso dovrete verificare che la piedinatura corrisponda. Per farlo, utilizzate una resistenza da 1000 ohm in serie ed un alimentatore da 5 volt. Collegate la resistenza al polo positivo; portate quindi il polo negativo a quello che ritenete possa essere il catodo comune. Provate ora a toccare con il terminale rimasto libero della resistenza gli altri piedini: dovrete vedere accendersi un segmento. Se ciò non accade, il catodo non è collegato al negativo: provate quindi a collegarlo agli altri piedini finchè un segmento non si accende. Battete in questo modo tutti i piedini e prendete nota di quale piedino fa accendere ogni segmento; notate come nel nostro display il catodo sia collegato a due piedini contemporaneamente. Se anche il vostro display è piedinato come il TFK514 potrete montarlo come segue, altrimenti dovrete inserire le resistenze in modo da adeguarvi alla piedinatura del vostro display.



Elenco componenti:

-) C1 100nF
-) R1...R7 390 ohm
-) IC1 Nutchip
-) IC2 4511
-) DISP1: TFK214 (catodo comune)
-) OSC1 risuonatore ceramico 4MHz 3 piedini

il circuito montato su una basetta sperimentale a molla

La tavola della verità

La tavola di un contatore è quanto di più semplice si possa immaginare. Si tratta di mettere sulle uscite, a cadenza di un secondo, tutti i valori binari corrispondenti alle cifre da 0 a 9. Troviamo questi valori nella tabella di pilotaggio dell'integrato 4511, basterà metterli negli stati da st00 a st09 ed il gioco è fatto. Ma attenzione: siccome per Nutstation l'uscita 1 è a sinistra mentre nei numeri binari si comincia da destra, i numeri binari appariranno rovesciati in Nutstation. Ecco perchè il numero 3 che è 0011 in binario diventa un 1100 sulle uscite dello stato st03.

Il file si chiama *counter.nut* :

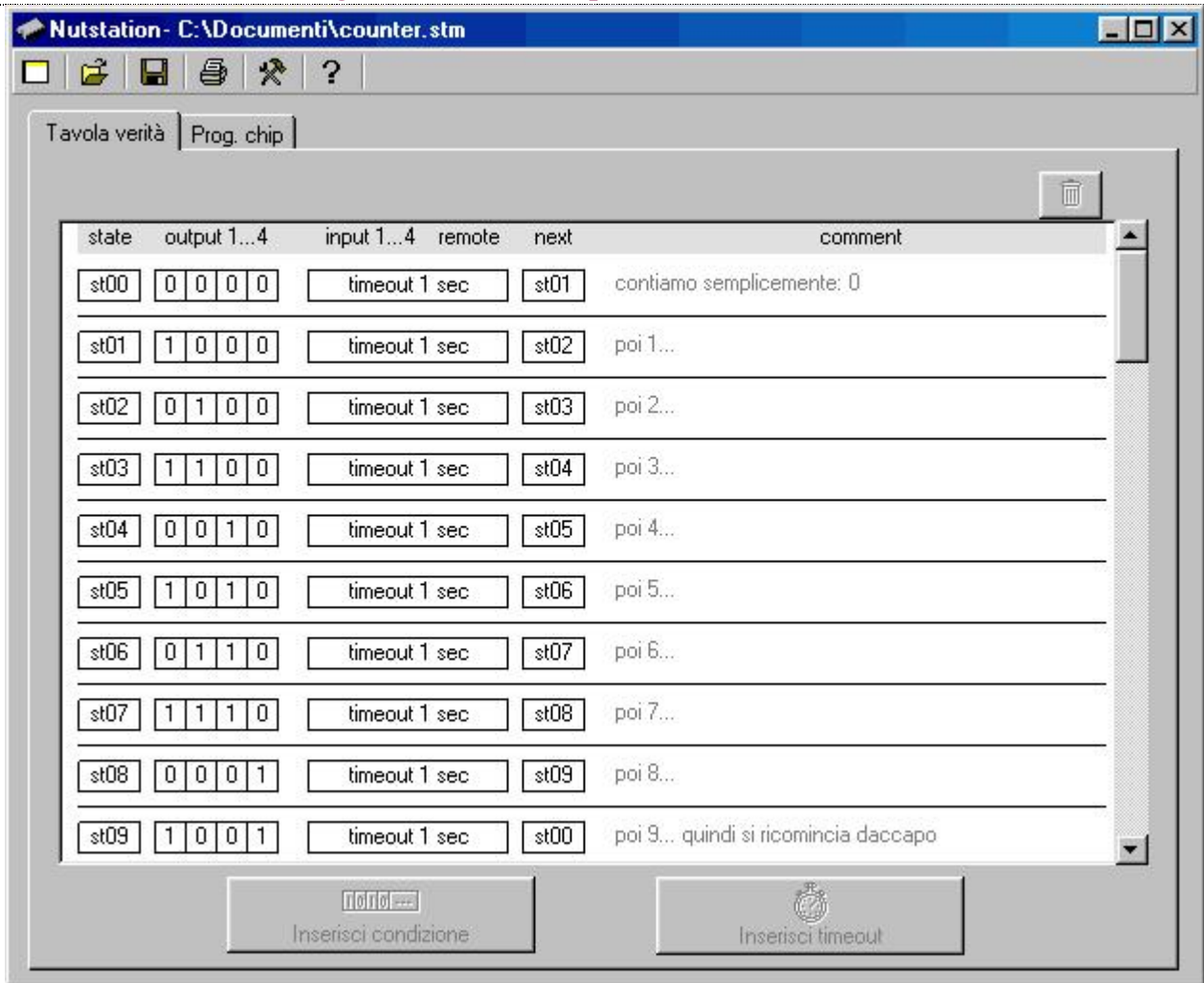


Tavola della verità per il conteggio in avanti.

Sareste capaci di....

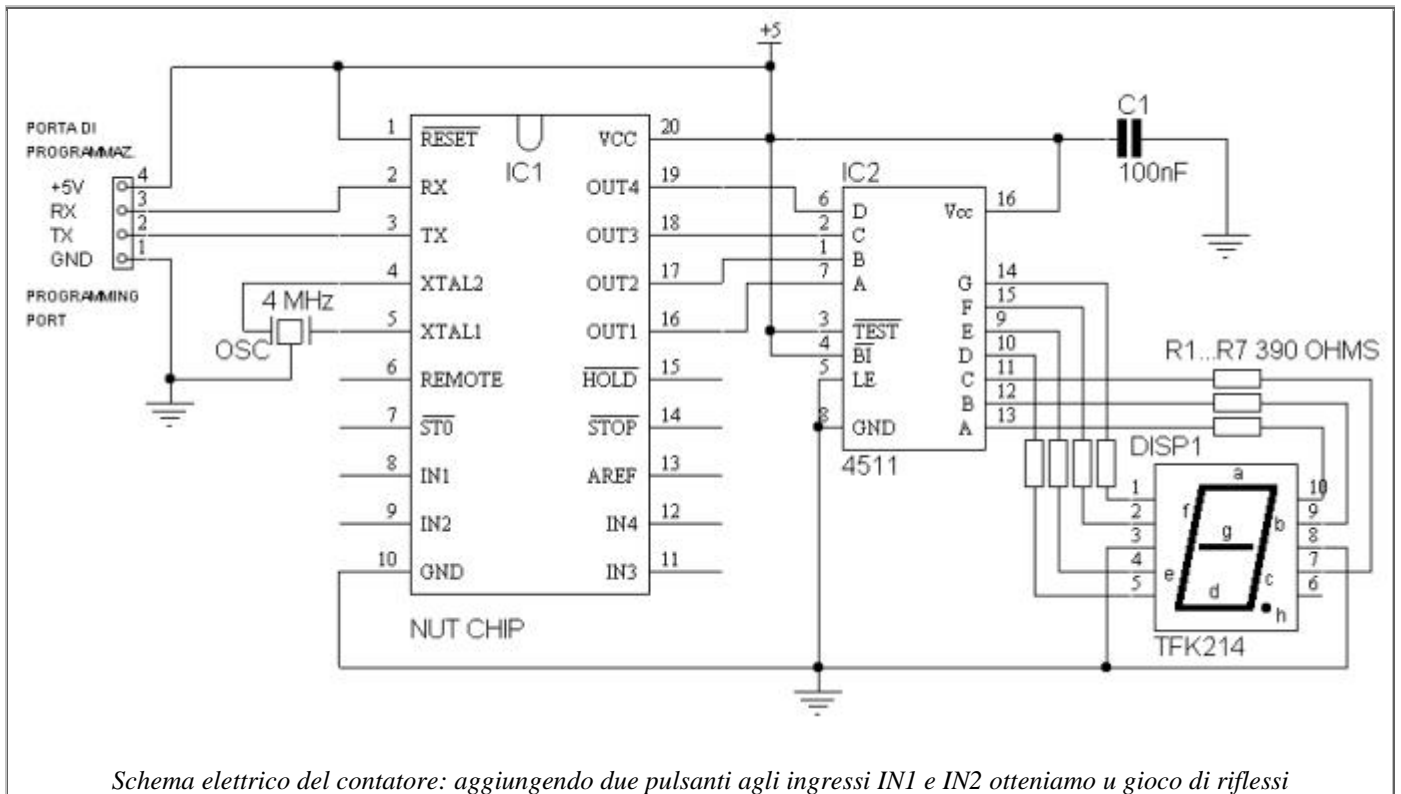
Una volta imparata la tecnica per realizzare un contatore e pilotare un display, si possono immaginare molti circuiti diversi
Vediamone alcuni:.

-)] Sareste capaci di progettare un **conto alla rovescia**? Potreste usarlo al prossimo capodanno o per lanciare il modellino di un missile.
-)] Sareste capaci di realizzare un [misuratore di riflessi](#)? Dovrete aggiungere un pulsante per fermare il conteggio. Più tardi lo fermate, meno punti incasserete!!!
-)] Sareste capaci di realizzare una versione tutta digitale del [dado elettronico](#)?
Se guardate bene, ci sono delle somiglianze. Naturalmente dovrete aggiungere un pulsante all'ingresso di -STOP oppure -HOLD. Il numero estratto compare sul display invece che con i LED disposti a faccia di dado come nel progetto originale.
-)] Sareste capaci di progettare un **indicatore di marcia per il motorino**? Dovrete usare due contatti per rilevare quando si cambia marcia. Inoltre la sequenza delle marce non parte da zero, per via del particolare meccanismo che ingrana la "prima".

Contatore

[Il display a 7 segmenti](#) | [Il pilota display 4511](#) | [Realizzazione](#) | [Tavola della verità](#) | [Come un dado](#)

Questo progetto introduce una funzione fondamentale dell'elettronica digitale: il *conteggio*. Realizzando un semplice contatore che avanza automaticamente a passi di un secondo impareremo come sono fatti e come funzionano i display a sette segmenti come vanno pilotati. Chi sapesse già tutto dell'argomento, può saltare direttamente alla seconda parte di questo progetto, in cui si sfrutta il contatore per realizzare un [gioco di riflessi](#).



Il display a 7 segmenti.

Un display a sette segmenti è costruito riunendo in un unico contenitore 7 led. Ogni led è sagomato in modo da illuminare un "segmento": illuminando due o più segmenti assieme potremo leggere sul display tutte le cifre da 0 a 9.

I segmenti sono indicati convenzionalmente con le lettere da A fino a G: potete vedere la disposizione dei segmenti sul display dallo schema elettrico.

Per esempio, il numero due si ottiene illuminando i segmenti A, B, G, E, D, mentre un tre si ottiene illuminando i segmenti A, B, C, D, G; per avere un otto invece accenderemotutti i segmenti da A a G.

A dispetto del nome, tutti i display oggi hanno un ottavo led che illumina il puntino decimale: questo segmento è indicato di solito dalla lettera H o dalla sigle DP (decimal point).

Per ridurre il numero dei piedini, all'interno del display i catodi di tutti i led sono collegati assieme e portati all'esterno su di un unico pin; sugli altri pin troveremo invece tutti gli anodi dei led da A fino ad H.. Questi display si chiamano "a catodo comune". Il display TFK214 utilizzato nel nostro progetto è proprio di questo tipo. Potete sostituirlo con qualsiasi altro display a sette segmenti purchè sia sempre a catodo comune: in tal caso però dovrete rifare quasi sicuramente i collegamenti perchè la piedinatura di solito non corrisponde da un tipo di display ad un altro. Ecco come è collegato il TFK214:

piedino	segmento
1	G
2	F
3	catodo comune
4	E
5	D
6	H (DP)
7	C
8	catodo comune
9	B
10	A

pinout del display TFK214

Il pilota display 4511

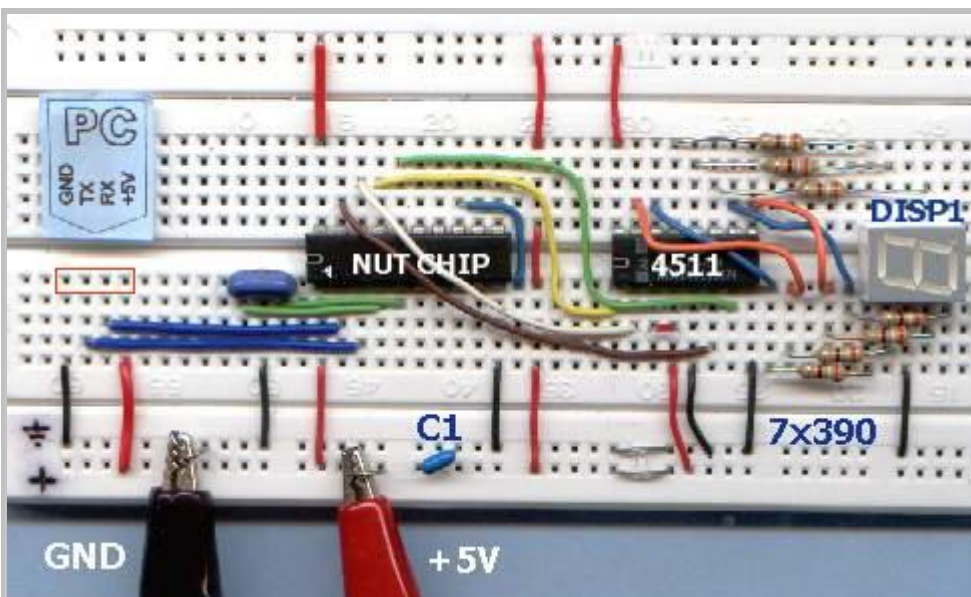
Il circuito integrato CD4511 è molto diffuso ed economico. La sua funzione è quella di "leggere" un numero binario sui suoi 4 ingressi A-B-C-D e di eccitare le uscite A-B-C-D-E-F-G in modo da accendere un display a catodo comune formando la cifra corrispondente. Come il NUT, anche il CD4511 ha al suo interno una tavola della verità, eccola:

numero	binario	segmenti	numero	binario	segmenti
0	0000	A,B,C,D,E,F	8	1000	A,B,C,D,E,F,G
1	0001	B,C	9	1001	A,B,C,F,G
2	0010	A,B,G,E,D	10	1010	A,B,C,D,E,F
3	0011	A,B,C,D,G	11	1011	spento
4	0100	F,G,B,C	12	1100	spento
5	0101	A,F,G,C,D	13	1101	spento
6	0110	F,G,C,D,E	14	1110	spento
7	0111	A,B,C	15	1111	spento

Il 4511 decodifica il numero binario sugli ingressi ed accende i segmenti corrispondenti

Realizzazione

Il montaggio non dovrebbe presentare alcuna difficoltà. E' però necessaria molta cura per non invertire i componenti polarizzati (display, NUTCHIP e 4511). Se avete un altro display a catodo comune potete utilizzarlo al posto del TFK514; in tal caso dovrete verificare che la piedinatura corrisponda. Per farlo, utilizzate una resistenza da 1000 ohm in serie ed un alimentatore da 5 volt. Collegate la resistenza al polo positivo; portate quindi il polo negativo a quello che ritenete possa essere il catodo comune. Provate ora a toccare con il terminale rimasto libero della resistenza gli altri piedini: dovrete vedere accendersi un segmento. Se ciò non accade, il catodo non è collegato al negativo: provate quindi a collegarlo agli altri piedini finchè un segmento non si accende. Battete in questo modo tutti i piedini e prendete nota di quale piedino fa accendere ogni segmento; notate come nel nostro display il catodo sia collegato a due piedini contemporaneamente. Se anche il vostro display è piedinato come il TFK514 potrete montarlo come segue, altrimenti dovrete inserire le resistenze in modo da adeguarvi alla piedinatura del vostro display.



Elenco componenti:

-) C1 100nF
-) R1...R7 390 ohm
-) IC1 Nutchip
-) IC2 4511
-) DISP1: TFK214 (catodo comune)
-) OSC1 risuonatore ceramico 4MHz 3 piedini

il circuito montato su una basetta sperimentale a molla

La tavola della verità

La tavola di un contatore è quanto di più semplice si possa immaginare. Si tratta di mettere sulle uscite, a cadenza di un secondo, tutti i valori binari corrispondenti alle cifre da 0 a 9. Troviamo questi valori nella tabella di pilotaggio dell'integrato 4511, basterà metterli negli stati da st00 a st09 ed il gioco è fatto. Ma attenzione: siccome per Nutstation l'uscita 1 è a sinistra mentre nei numeri binari si comincia da destra, i numeri binari appariranno rovesciati in Nutstation. Ecco perchè il numero 3 che è 0011 in binario diventa un 1100 sulle uscite dello stato st03.

Il file si chiama *counter.nut* :

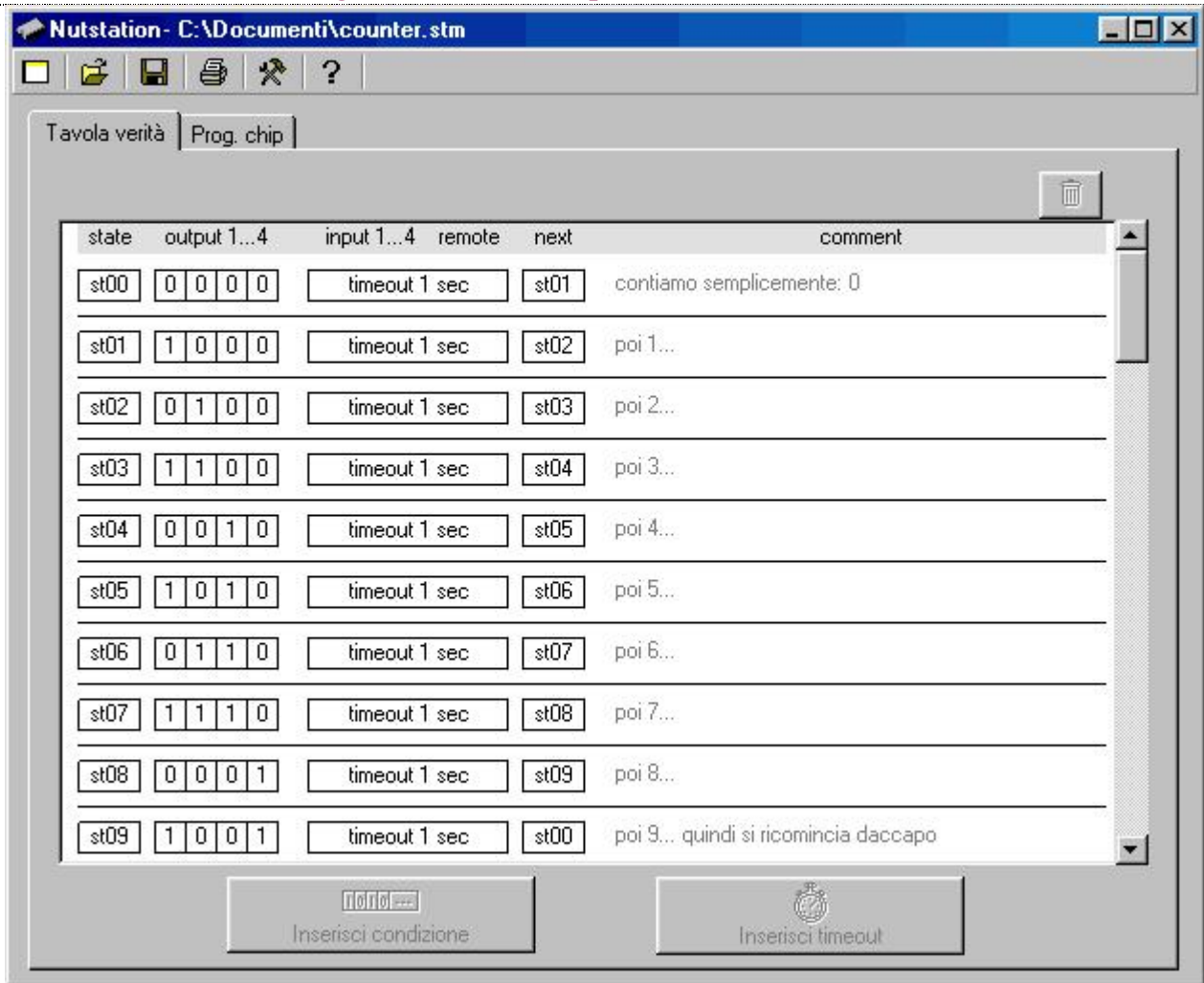


Tavola della verità per il conteggio in avanti.

Sareste capaci di....

Una volta imparata la tecnica per realizzare un contatore e pilotare un display, si possono immaginare molti circuiti diversi

Vediamone alcuni:.

-)] Sareste capaci di progettare un **conto alla rovescia**? Potreste usarlo al prossimo capodanno o per lanciare il modellino di un missile.
-)] Sareste capaci di realizzare un [misuratore di riflessi](#)? Dovrete aggiungere un pulsante per fermare il conteggio. Più tardi lo fermate, meno punti incasserete!!!
-)] Sareste capaci di realizzare una versione tutta digitale del [dado elettronico](#)? Se guardate bene, ci sono delle somiglianze. Naturalmente dovrete aggiungere un pulsante all'ingresso di -STOP oppure -HOLD. Il numero estratto compare sul display invece che con i LED disposti a faccia di dado come nel progetto originale.
-)] Sareste capaci di progettare un **indicatore di marcia per il motorino**? Dovrete usare due contatti per rilevare quando si cambia marcia. Inoltre la sequenza delle marce non parte da zero, per via del particolare meccanismo che ingrana la "prima".

Gioco di riflessi

Rivela anche le false partenze

[Questione di velocità](#) | [Schema elettrico](#) | [Funzionamento](#) | [Montaggio](#)

Una questione di velocità

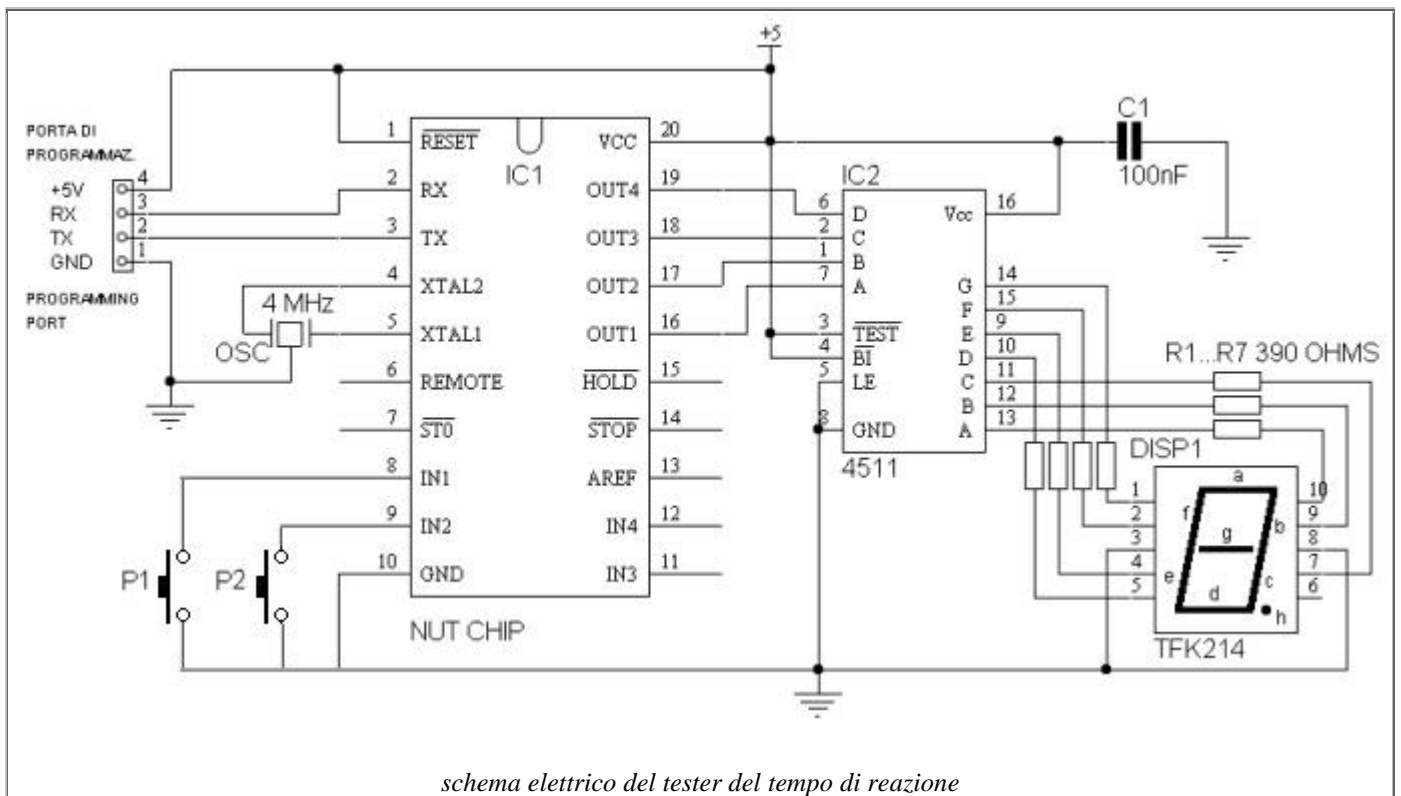
Silenzio. Il display è spento. Cinque lunghi secondi che sembrano non finire mai. All'improvviso tutti i segmenti si accendono, ed il tempo cominciano a scorrere inesorabile. Siamo noi che dobbiamo fermarlo, premendo un pulsante: il display ci mostra il nostro punteggio. Zero, velocità da tartaruga; sei, appena sufficiente; otto, riflessi attenti e pronti; nove, la prontezza di un pugile o di un centometrista!

Lo schema

Lo schema del gioco di riflessi è molto simile al progetto del [contatore](#) a display. Aggiungiamo a quel progetto due pulsanti, P1 e P2, collegati rispettivamente agli ingressi 1 e 2.

P2 serve a far partire il gioco. Quando lo si preme il display si spegne per riaccendersi dopo 5 secondi esatti: è il momento in cui il gioco prende vita!

P1 è il pulsante che dobbiamo premere, il più velocemente possibile, appena vediamo il display accendersi.



Funzionamento

Per capire meglio la tavola del gioco la possiamo dividere in due:

- J) la prima parte misura il tempo di reazione: sono gli stati da st00 a st09, che scorrono via via.
 - J) la seconda parte visualizza il punteggio: sono gli stati da st10 a st19, dove st10 = 0, st11=1, st12=2, eccetera.
- Questa parte è molto simile a quella del progetto di un [contatore](#) a display.

Partiamo naturalmente dallo stato st00. Qui si attendono 5 secondi a display spento (le combinazioni maggiori di nove spengono i display, ed 1111 in binario = 15 in decimale) per poi passare agli stati successivi. Se premiamo il pulsante prima che siano passati i 5 secondi allora si tratta di una falsa partenza: il punteggio è zero, saltiamo direttamente allo stato st10 che visualizza uno zero tondo tondo sul display.

Se abbiamo i nervi abbastanza saldi per evitare di partire in anticipo, dopo 5 secondi si passa allo stato st01. Qui il display si accende a piena luce, mostrando la cifra 8: non si tratta ancora del punteggio, abbiamo scelto questa cifra perchè è la più visibile avendo tutti i segmenti accesi. Si resta nello stato st01 per 100 mS appena. Se si preme il pulsante collegato all'ingresso 1 entro tale tempo, otteniamo il punteggio massimo, 9 punti. Premendo P1 quindi si passa allo stato st19.

Reagire nei primi 100 mS è impossibile, ma un un pò di fortuna ci si può riuscire (altrimenti potrete modificare il tempo aumentandolo a vostra discrezione). Se siete comuni mortali come noi, il tempo passerà prima che siate fisicamente in grado di premere il pulsante portandoci nello stato st02.

Lo stato st02 funziona in modo identico al precedente, con l'unica differenza che il tempo è un pò più breve (questo rende il gioco più interessante) e che se premiamo il pulsante facciamo otto punti: cioè finiamo nello stato st08.

Gli stati successivi fino a st09 si ripetono secondo il medesimo schema, uno dopo l'altro. Via via il punteggio decresce, fino ad un punto per lo stato st09. Se quando anche lo stato st09 finisce non doveste avere premuto nessun pulsante, il NUTCHIP vi classifica irrimediabilmente come lumache passando a visualizzare uno zero tondo tondo.

state	output 1...4	input 1...4	remote	next	comment
st00	1 1 1 1	0 - - - ...	timeout 5 sec	st10 st01	rivelazione falsa partenza, 0 punti display spento in attesa di partire
st01	0 0 0 1	0 - - - ...	timeout 100 mS	st19 st02	primo passo: 9 punti (il display mostra sempre 8 per farsi vedere bene)
st02	0 0 0 1	0 - - - ...	timeout 70 mS	st18 st03	secondo passo: 8 punti
st03	0 0 0 1	0 - - - ...	timeout 70 mS	st17 st04	7 punti
st04	0 0 0 1	0 - - - ...	timeout 50 mS	st16 st05	6 punti
st05	0 0 0 1	0 - - - ...	timeout 50 mS	st15 st06	5 punti
st06	0 0 0 1	0 - - - ...	timeout 70 mS	st14 st07	4 punti
st07	0 0 0 1	0 - - - ...	timeout 70 mS	st13 st08	3 punti
st08	0 0 0 1	0 - - - ...	timeout 100 mS	st12 st09	2 punti
st09	0 0 0 1	0 - - - ...	timeout 100 mS	st11 st10	1 punto fuori tempo massimo, zero punti

tavola del gioco, prima parte: misura del tempo di risposta

La seconda parte della tavola è un semplice codificatore per il display, ad ogni stato corrisponde un punteggio diverso. Così st10 è lo zero, st11 è l'uno, st12 il due e così via fino ad st19 che è il nove. Se avete seguito con attenzione l'illustrazione del [contatore](#) dovrete riconoscere le analogie con facilità.

Il punteggio resta visualizzato fino a che non si preme il tasto P2: a quel punto il gioco ricomincia, con altri cinque lunghissimi secondi d'attesa.

Il modo migliore per prendere confidenza con questo progetto è realizzarlo. Non solo sarete in grado di capire il funzionamento dettagliato di ogni parte, sarete anche in grado di eseguire quelle modifiche per adattarlo ai vostri gusti. Impiegando il 77% delle risorse, si tratta di un

st10	0 0 0 0	- 0 - - ...	st00
st11	1 0 0 0	- 0 - - ...	st00
st12	0 1 0 0	- 0 - - ...	st00
st13	1 1 0 0	- 0 - - ...	st00
st14	0 0 1 0	- 0 - - ...	st00
st15	1 0 1 0	- 0 - - ...	st00
st16	0 1 1 0	- 0 - - ...	st00
st17	1 1 1 0	- 0 - - ...	st00
st18	0 0 0 1	- 0 - - ...	st00
st19	1 0 0 1	- 0 - - ...	st00

progetto di complessità medio alta che illustra come misurare dei tempi. e come verificare che rientrino entro valori prestabiliti: un'esigenza particolarmente sentita in molti settori dell'automazione.

Montaggio

Il montaggio può avvenire su una basetta sperimentale a molla come quella utilizzata nel progetto del [contatore](#), per cui vi rimandiamo a tale progetto per la foto del prototipo. I componenti sono disposti allo stesso modo, con l'aggiunta di P1 e P2 collegati dagli ingressi IN1 e IN2 verso GND.

seconda parte, visualizzazione del punteggio: la tavola completa si trova nel file "react.nut"

Elenco componenti:

-) C1 100nF
-) R1...R7 390 ohm
-) IC1 Nutchip
-) IC2 4511
-) DISP1: TFK214 (catodo comune)
-) OSC1 risuonatore ceramico 4MHz 3 piedini
-) P1, P2: pulsanti normalmente aperti

Gioco di riflessi

Rivela anche le false partenze

[Questione di velocità](#) | [Schema elettrico](#) | [Funzionamento](#) | [Montaggio](#)

Una questione di velocità

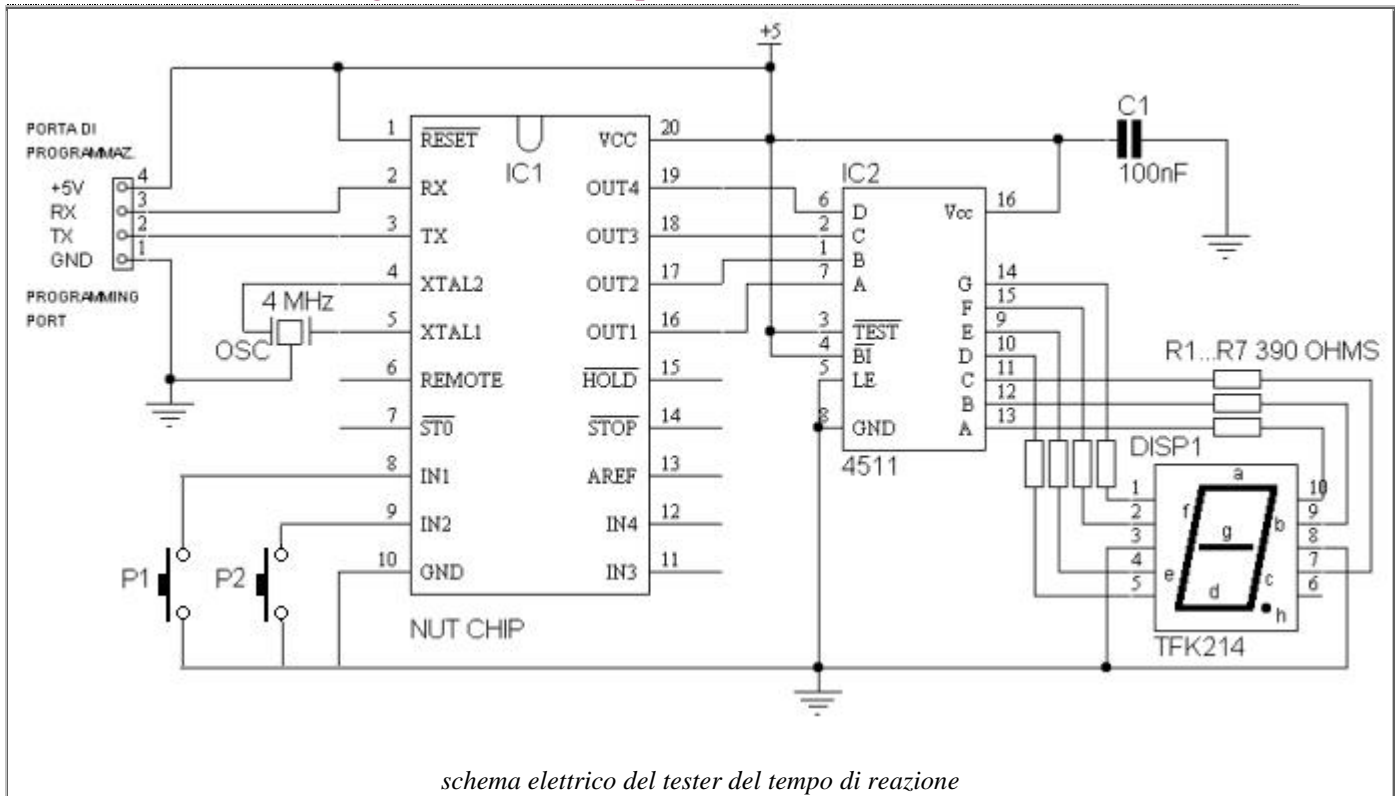
Silenzio. Il display è spento. Cinque lunghi secondi che sembrano non finire mai. All'improvviso tutti i segmenti si accendono, ed il tempo cominciano a scorrere inesorabile. Siamo noi che dobbiamo fermarlo, premendo un pulsante: il display ci mostra il nostro punteggio. Zero, velocità da tartaruga; sei, appena sufficiente; otto, riflessi attenti e pronti; nove, la prontezza di un pugile o di un centometrista!

Lo schema

Lo schema del gioco di riflessi è molto simile al progetto del [contatore](#) a display. Aggiungiamo a quel progetto due pulsanti, P1 e P2, collegati rispettivamente agli ingressi 1 e 2.

P2 serve a far partire il gioco. Quando lo si preme il display si spegne per riaccendersi dopo 5 secondi esatti: è il momento in cui il gioco prende vita!

P1 è il pulsante che dobbiamo premere, il più velocemente possibile, appena vediamo il display accendersi.



Funzionamento

Per capire meglio la tavola del gioco la possiamo dividere in due:

-)] la prima parte misura il tempo di reazione: sono gli stati da st00 a st09, che scorrono via via.
-)] la seconda parte visualizza il punteggio: sono gli stati da st10 a st19, dove st10 = 0, st11=1, st12=2, eccetera.

Questa parte è molto simile a quella del progetto di un [contatore](#) a display.

Partiamo naturalmente dallo stato st00. Qui si attendono 5 secondi a display spento (le combinazioni maggiori di nove spengono i display, ed 1111 in binario = 15 in decimale) per poi passare agli stati successivi. Se premiamo il pulsante prima che siano passati i 5 secondi allora si tratta di una falsa partenza: il punteggio è zero, saltiamo direttamente allo stato st10 che visualizza uno zero tondo tondo sul display.

Se abbiamo i nervi abbastanza saldi per evitare di partire in anticipo, dopo 5 secondi si passa allo stato st01. Qui il display si accende a piena luce, mostrando la cifra 8: non si tratta ancora del punteggio, abbiamo scelto questa cifra perchè è la più visibile avendo tutti i segmenti accesi. Si resta nello stato st01 per 100 mS appena. Se si preme il pulsante collegato all'ingresso 1 entro tale tempo, otteniamo il punteggio massimo, 9 punti. Premendo P1 quindi si passa allo stato st19.

Reagire nei primi 100 mS è impossibile, ma un un pò di fortuna ci si può riuscire (altrimenti potrete modificare il tempo aumentandolo a vostra discrezione). Se siete comuni mortali come noi, il tempo passerà prima che siate fisicamente in grado di premere il pulsante portandoci nello stato st02.

Lo stato st02 funziona in modo identico al precedente, con l'unica differenza che il tempo è un pò più breve (questo rende il gioco più interessante) e che se premiamo il pulsante facciamo otto punti: cioè finiamo nello stato st08.

Gli stati successivi fino a st09 si ripetono secondo il medesimo schema, uno dopo l'altro. Via via

il punteggio decresce, fino ad un punto per lo stato st09. Se quando anche lo stato st09 finisce non dovete avere premuto nessun pulsante, il NUTCHIP vi classifica irrimediabilmente come lumache passando a visualizzare uno zero tondo tondo.

state	output 1...4	input 1...4	remote	next	comment
st00	1 1 1 1	0 - - - ... timeout 5 sec		st10 st01	rivelazione falsa partenza, 0 punti display spento in attesa di partire
st01	0 0 0 1	0 - - - ... timeout 100 mS		st19 st02	primo passo: 9 punti (il display mostra sempre 8 per farsi vedere bene)
st02	0 0 0 1	0 - - - ... timeout 70 mS		st18 st03	secondo passo: 8 punti
st03	0 0 0 1	0 - - - ... timeout 70 mS		st17 st04	7 punti
st04	0 0 0 1	0 - - - ... timeout 50 mS		st16 st05	6 punti
st05	0 0 0 1	0 - - - ... timeout 50 mS		st15 st06	5 punti
st06	0 0 0 1	0 - - - ... timeout 70 mS		st14 st07	4 punti
st07	0 0 0 1	0 - - - ... timeout 70 mS		st13 st08	3 punti
st08	0 0 0 1	0 - - - ... timeout 100 mS		st12 st09	2 punti
st09	0 0 0 1	0 - - - ... timeout 100 mS		st11 st10	1 punto fuori tempo massimo, zero punti

tavola del gioco, prima parte: misura del tempo di risposta

La seconda parte della tavola è un semplice codificatore per il display, ad ogni stato corrisponde un punteggio diverso. Così st10 è lo zero, st11 è l'uno, st12 il due e così via fino ad st19 che è il nove. Se avete seguito con attenzione l'illustrazione del [contatore](#) dovrete riconoscere le analogie con facilità.

Il punteggio resta visualizzato fino a che non si preme il tasto P2: a quel punto il gioco ricomincia, con altri cinque lunghissimi secondi d'attesa.

Il modo migliore per prendere confidenza con questo progetto è realizzarlo. Non solo sarete in grado di capire il funzionamento dettagliato di ogni parte, sarete anche in grado di eseguire quelle modifiche per adattarlo ai vostri gusti. Impiegando il 77% delle risorse, si tratta di un progetto di complessità medio alta che illustra come misurare dei tempi, e come verificare che rientrino entro valori prestabiliti: un'esigenza particolarmente sentita in molti settori dell'automazione.

Montaggio

Il montaggio può avvenire su una basetta sperimentale a molla come quella utilizzata nel progetto del [contatore](#), per cui vi rimandiamo a tale progetto per la foto del prototipo. I componenti sono disposti allo stesso modo, con l'aggiunta di P1 e P2 collegati dagli ingressi IN1 e IN2 verso GND.

st10	0 0 0 0	- 0 - - ...	st00
st11	1 0 0 0	- 0 - - ...	st00
st12	0 1 0 0	- 0 - - ...	st00
st13	1 1 0 0	- 0 - - ...	st00
st14	0 0 1 0	- 0 - - ...	st00
st15	1 0 1 0	- 0 - - ...	st00
st16	0 1 1 0	- 0 - - ...	st00
st17	1 1 1 0	- 0 - - ...	st00
st18	0 0 0 1	- 0 - - ...	st00
st19	1 0 0 1	- 0 - - ...	st00

seconda parte, visualizzazione del punteggio: la tavola completa si trova nel file "react.nut"

Elenco componenti:

-) C1 100nF
-) R1...R7 390 ohm
-) IC1 Nutchip
-) IC2 4511
-) DISP1: TFK214 (catodo comune)
-) OSC1 risuonatore ceramico 4MHz 3 piedini
-) P1, P2: pulsanti normalmente aperti

Sperimentare con un encoder

[Potenziometri "digitali"](#) | [Schema elettrico](#) | [Montaggio](#) | [Rilevamento](#) | [LED rotanti](#) | [LED a barra](#) | [LED a sequenza](#)

Potenziometri "digitali"

I dispositivi più **moderni** utilizzano gli encoder sempre più **spesso**. I vecchi **potenziometri** e **commutatori** vengono via via rimpiazzati da questi dispositivi digitali, semplici e misteriosi al tempo stesso. La funzione di un encoder è semplice: convertire in impulsi digitali la rotazione di un albero. L'albero in questione può essere molto differente da un'applicazione all'altra....

-] il pomello del volume di un'autoradio
-] il meccanismo principale di una macchina automatica grande quanto una rotativa:
-] la manopola del timer di un forno a microonde
-] la base dei tempi di un oscilloscopio
-] la temperatura del condizionatore dell'auto

pur se con dimensioni, caratteristiche elettriche, robustezza meccanica, velocità e precisione differenti, ruotando il perno degli encoder incrementali otteniamo sempre la medesima sequenza di impulsi.

Esternamente gli encoder si presentano come un blocchetto plastico o metallico. Da una delle pareti fuoriesce un perno, da un'altra invece tre piedini. Uno di questi piedini è il comune, dagli altri due invece fuoriescono gli impulsi quando si ruota il perno. Prenderemo in considerazione i **tipi più economici di encoder**, quelli in cui i **contatti sono costituiti da semplici interruttori** (sulle macchine automatiche si montano invece spesso gli encoder di tipo **ottico**, dove gli interruttori sono stati sostituiti da delle coppie di led e fototransistor).

Il modello utilizzato nei nostri esperimenti è di produzione **ALPS**, e produce **24 impulsi per giro**, ad ogni impulso si avverte uno "**scatto**" sul perno. Si utilizza in prodotti di largo consumo (con 24 scatti è l'ideale per gli orologi!), ed in commercio se ne possono trovare di simili a 36 ed anche 48 impulsi per giro.

Se avete la fortuna di possedere un encoder del tipo usato nelle macchine automatiche, noterete che questi hanno un **movimento continuo**.

Questi encoder sono quasi sempre ottici (in un modo simile ad un mouse), ed hanno una risoluzione di 256 impulsi per giro ed anche più. Troverete allora dei piedini supplementari che servono ad alimentare l'elettronica interna. Chi possiede questi encoder sa anche come collegarli quindi non ci dilungheremo oltre.

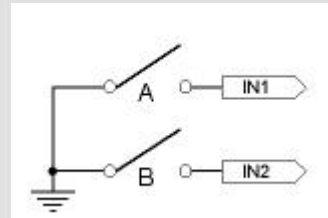
In questo esperimento vedremo:

-] come si collega un encoder al NUT
-] come si interpretano gli impulsi per

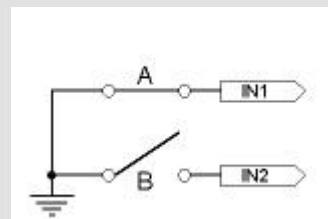
Quattro passi assieme all'encoder

Ruotando la manopola di un encoder, per ognuno degli scatti che avvertiamo sul pomello, sui suoi piedini si sussegue una precisa serie di impulsi.

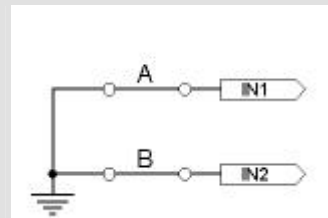
Gli encoder relativi sono costituiti da una coppia di contatti: esternamente hanno solo tre piedini perchè due di essi sono riuniti assieme. Con la manopola ferma su uno degli scatti, entrambi i contatti A e B sono aperti. Appena si ruota la manopola i contatti si chiudono secondo una sequenza ben precisa...



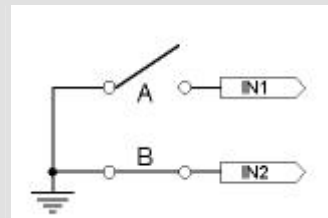
Se ruotiamo la manopola in senso **orario**, il **primo** a chiudersi sarà il contatto A.



Continuando la ruotare la manopola, quando questa sarà a **metà strada** fra uno scatto e l'altro troveremo **entrambi** i contatti chiusi.

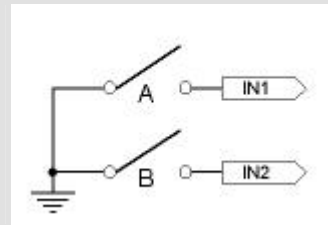


Prima che il movimento si concluda, il contatto A si **riaprirà** lasciando chiuso il **solo** contatto B.



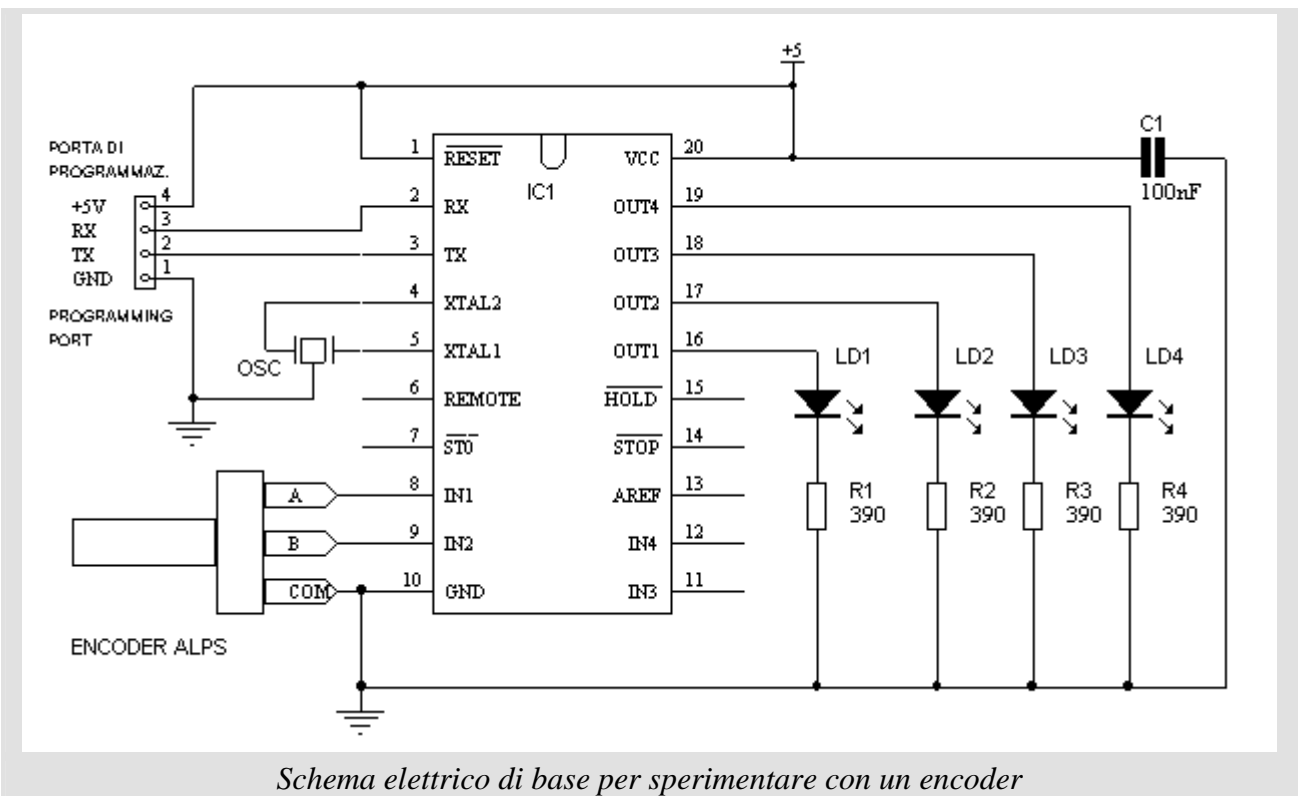
fare ruotare la luce di un LED

Quando la manopola raggiungerà lo **scatto**, entrambi i contatti saranno di nuovo **riaperti**. La sequenza si ripete identica per tutti gli scatti successivi. Se invece si ruota la manopola in senso **antiorario**, la sequenza avviene al **contrario**.



Schema elettrico

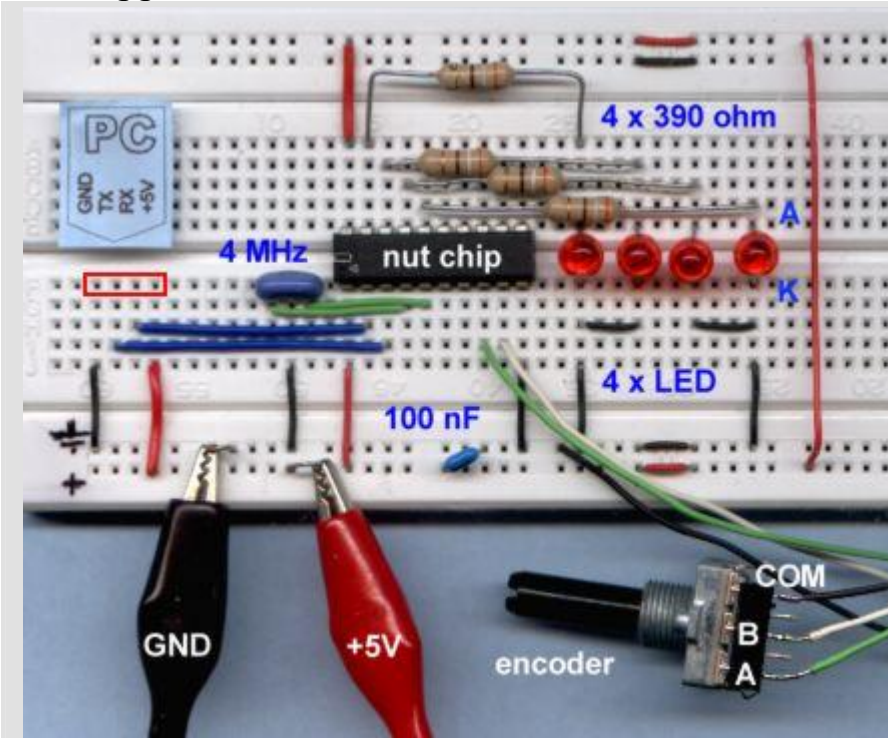
Gli encoder meccanici sono del tutto simili ad una coppia di contatti. E' come se dentro ad ogni encoder ci fossero due pulsanti (A e B) che si pigiano con una sequenza prestabilita ogni volta che ne ruotiamo l'alberino. I pulsanti si collegano direttamente fra gli ingressi del NUTCHIP e GND: collegheremo quindi A all'ingresso 1, B all'ingresso 2, ed il terminale comune (COM) al negativo (GND).



Schema elettrico di base per sperimentare con un encoder

Alle uscite colleghiamo 4 LED, ponendo in serie ad ognuno una resistenza per limitare la corrente massima.

Montaggio



Componenti usati:

- 4 x resistenze da 390 ohm
- 4 x LED
- condensatore da 100nF
- encoder ALPS
- oscillatore ceramico 4 MHz (a tre piedini)
- Nutchip

La fotografia mostra come montare il circuito sulla basetta sperimentale. Fate attenzione a non dimenticare nessuno dei numerosi ponticelli di filo presenti. Il katodo (K) dei LED è quello dalla parte smussata della capsula, e corrisponde anche al terminale più corto.

Potete saldare degli spezzi di filo ai piedini dell'encoder, oppure potete costruirvi un zoccolo speciale tagliando un pezzo di un comune zoccolo per integrati. Il passo fra un piedino e l'altro dell'encoder (5,08 mm) infatti è esattamente il doppio del passo degli integrati (2,54 mm).



Rilevare il movimento

Rilevare il movimento di un encoder è facile: basta attendere che si chiudano i suoi contatti. Dal tipo di sequenza che si presenta si deduce poi il senso di rotazione della manopola. Nel nostro schema, A è collegato all'ingresso IN1 e B ad IN2. Quando il contatto è aperto, il relativo ingresso è a 1; quando il contatto si chiude sull'ingresso si leggerà invece uno 0.

movimento ORARIO		movimento ANTIORARIO	
posizione di riposo	IN1 = 1, IN2 = 1	posizione di riposo	IN1 = 1, IN2 = 1
inizio movimento orario (A)	IN1 = 0, IN2 = 1	inizio movimento orario (B)	IN1 = 1, IN2 = 0
metà movimento (A+B)	IN1 = 0, IN2 = 0	metà movimento (A+B)	IN1 = 0, IN2 = 0

fine movimento orario (B)	IN1 = 1, IN2 = 0	fine movimento orario (A)	IN1 = 0, IN2 = 1
---------------------------	---------------------	---------------------------	---------------------

La tabella del Nutchip collegherà uno stato all'altro tenendo traccia di queste fasi. La chiusura di entrambi i contatti indicherà il momento di partire: ci porteremo in uno stato intermedio, attendendo che la riapertura di uno dei due ci indichi il verso di rotazione. Potremo quindi portarci definitivamente nello stato successivo, dove potremo eventualmente attendere che si riaprano entrambi i contatti (se temiamo che il nostro encoder abbia dei rimbalzi).

LED rotanti

Il primo esperimento è quello di realizzare delle semplici luci rotanti. Girando l'encoder, il LED acceso si muoverà da destra a sinistra o da sinistra a destra. Questo tipo di funzionamento è analogo a quello di un SELETTORE a quattro vie. Quando si avanza oltre l'ultimo LED si riaccende il primo, e viceversa. Il file da caricare è *encoder_rot.nut*.

LED a barra

In questo esperimento i LED si accendono progressivamente. Questo tipo di visualizzazione dà l'impressione di una grandezza che aumenta o diminuisce ed è adatto per esempio ad un controllo di volume o di velocità di un motore.

A differenza del controllo precedente, quando si avanza dopo che anche l'ultimo LED si è acceso, la sequenza non ricomincia da capo ma resta come congelata. Il file da caricare è *encoder_bar.nut*

LED a sequenza predefinita

Negli esperimenti precedenti abbiamo disposto le uscite in modo da dare un'accensione "ordinata" ai LED, uno dopo l'altro. In realtà possiamo scrivere nelle uscite una qualsiasi altra sequenza: ad esempio, se usiamo dei LED rossi, gialli e verdi (ed eventualmente un LED tverde a forma di freccia) possiamo simulare la sequenza di un semaforo (solo rosso; verde+freccia; verde+giallo+freccia; rosso+freccia...).

Contatore binario

Un caso speciale di uscite in sequenza predefinita è dato dal contatore binario. Per un contatore binario, le uscite si attivano in una sequenza che rispecchia quella delle cifre dei numeri binari. Una tale sequenza si può usare per pilotare ad esempio un display.

I più esperti avranno già capito come farlo... Potete vederlo nel prossimo progetto, un [display regolabile ad encoder](#).

Sperimentare con un encoder

[Potenziometri "digitali"](#) | [Schema elettrico](#) | [Montaggio](#) | [Rilevamento](#) | [LED rotanti](#) | [LED a barra](#) | [LED a sequenza](#)

Potenziometri "digitali"

I dispositivi più **moderni** utilizzano gli encoder sempre più **spesso**. I vecchi **potenziometri e commutatori** vengono via via rimpiazzati da questi dispositivi digitali, semplici e misteriosi al tempo stesso. La funzione di un encoder è semplice: convertire in impulsi digitali la rotazione di un albero. L'albero in questione può essere molto differente da un'applicazione all'altra....

- J il pomello del volume di un'autoradio
- J il meccanismo principale di una macchina automatica grande quanto una rotativa:
 - J la manopola del timer di un forno a microonde
 - J la base dei tempi di un oscilloscopio
 - J la temperatura del condizionatore dell'auto

pur se con dimensioni, caratteristiche elettriche, robustezza meccanica, velocità e precisione differenti, ruotando il perno degli encoder incrementali otteniamo sempre la medesima sequenza di impulsi.

Esternamente gli encoder si presentano come un blocchetto plastico o metallico. Da una delle pareti fuoriesce un perno, da un'altra invece tre piedini. Uno di questi piedini è il comune, dagli altri due invece fuoriescono gli impulsi quando si ruota il perno. Prenderemo in considerazione i **tipi più economici di encoder**, quelli in cui i **contatti sono costituiti da semplici interruttori** (sulle macchine automatiche si montano invece spesso gli encoder di tipo **ottico**, dove gli interruttori sono stati sostituiti da delle coppie di led e fototransistor).

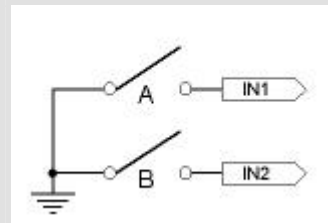
Il modello utilizzato nei nostri esperimenti è di produzione **ALPS**, e produce **24 impulsi per giro**, ad ogni impulso si avverte uno "**scatto**" sul perno. Si utilizza in prodotti di largo consumo (con 24 scatti è l'ideale per gli orologi!), ed in commercio se ne possono trovare di simili a 36 ed anche 48 impulsi per giro.

Se avete la fortuna di possedere un encoder del tipo usato nelle macchine automatiche, noterete che questi hanno un **movimento continuo**. Questi encoder sono quasi sempre ottici (in un modo simile ad un mouse), ed hanno una risoluzione di 256 impulsi per giro ed anche più. Troverete allora dei piedini supplementari che servono ad alimentare l'elettronica interna. Chi

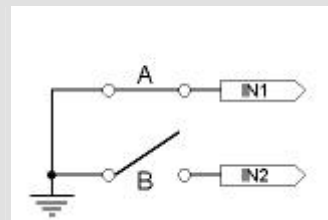
Quattro passi assieme all'encoder

Ruotando la manopola di un encoder, per ognuno degli scatti che avvertiamo sul pomello, sui suoi piedini si sussegue una precisa serie di impulsi.

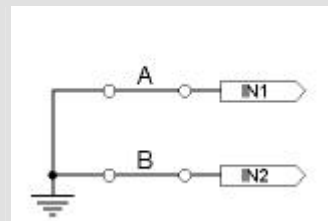
Gli encoder relativi sono costituiti da una coppia di contatti: esternamente hanno solo tre piedini perchè due di essi sono riuniti assieme. Con la manopola ferma su uno degli scatti, entrambi i contatti A e B sono aperti. Appena si ruota la manopola i contatti si chiudono secondo una sequenza ben precisa...



Se ruotiamo la manopola in senso **orario**, il **primo** a chiudersi sarà il contatto A.



Continuando la ruotare la manopola, quando questa sarà a **metà strada** fra uno scatto e l'altro troveremo **entrambi** i contatti chiusi.

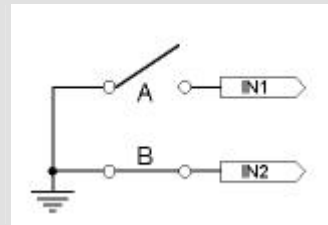


possiede questi encoder sa anche come collegarli quindi non ci dilungheremo oltre.

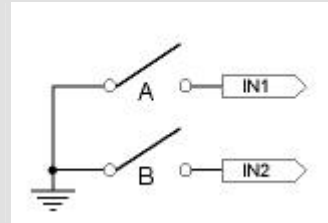
In questo esperimento vedremo:

-) come si collega un encoder al NUT
-) come si interpretano gli impulsi per fare ruotare la luce di un LED

Prima che il movimento si concluda, il contatto A si **riaprirà** lasciando chiuso il **solo** contatto B.

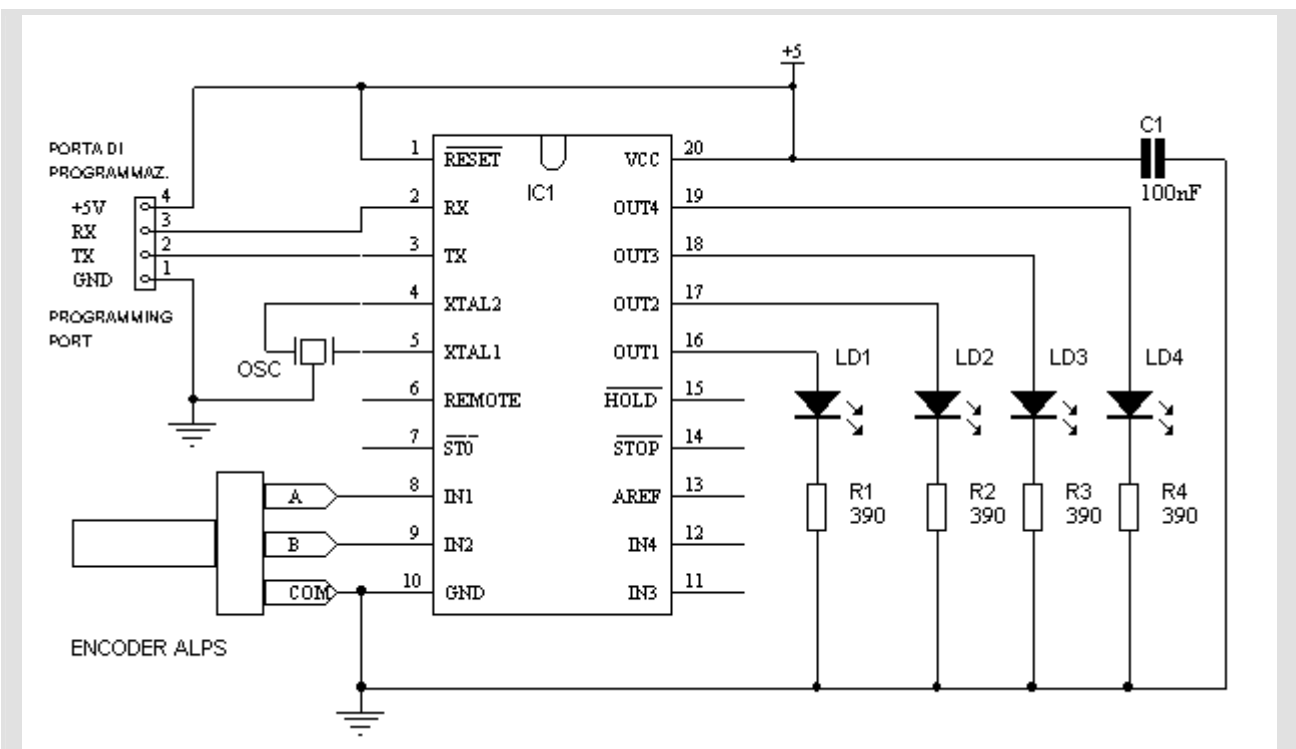


Quando la manopola raggiungerà lo **scatto**, entrambi i contatti saranno di nuovo **riaperti**. La sequenza si ripete identica per tutti gli scatti successivi. Se invece si ruota la manopola in senso **antiorario**, la sequenza avviene al **contrario**.



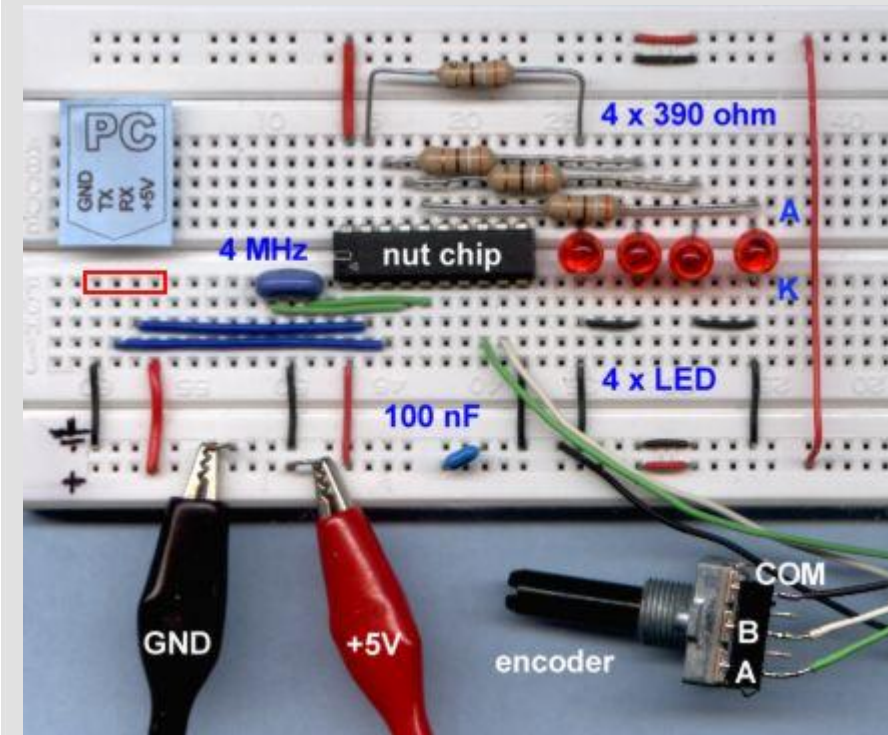
Schema elettrico

Gli encoder meccanici sono del tutto simili ad una coppia di contatti. E' come se dentro ad ogni encoder ci fossero due pulsanti (A e B) che si pigiano con una sequenza prestabilita ogni volta che ne ruotiamo l'alberino. I pulsanti si collegano direttamente fra gli ingressi del NUTCHIP e GND: collegheremo quindi A all'ingresso 1, B all'ingresso 2, ed il terminale comune (COM) al negativo (GND).



Alle uscite colleghiamo 4 LED, ponendo in serie ad ognuno una resistenza per limitare la corrente massima.

Montaggio



Componenti usati:

- 4 x resistenze da 390 ohm
- 4 x LED
- condensatore da 100nF
- encoder ALPS
- oscillatore ceramico 4 MHz (a tre piedini)
- Nutchip

La fotografia mostra come montare il circuito sulla basetta sperimentale. Fate attenzione a non dimenticare nessuno dei numerosi ponticelli di filo presenti. Il katodo (K) dei LED è quello dalla parte smussata della capsula, e corrisponde anche al terminale più corto.

Potete saldare degli spezzi di filo ai piedini dell'encoder, oppure potete costruirvi un zoccolo speciale tagliando un pezzo di un comune zoccolo per integrati. Il passo fra un piedino e l'altro dell'encoder (5,08 mm) infatti è esattamente il doppio del passo degli integrati (2,54 mm).



Rilevare il movimento

Rilevare il movimento di un encoder è facile: basta attendere che si chiudano i suoi contatti. Dal tipo di sequenza che si presenta si deduce poi il senso di rotazione della manopola. Nel nostro schema, A è collegato all'ingresso IN1 e B ad IN2. Quando il contatto è aperto, il relativo ingresso è a 1; quando il contatto si chiude sull'ingresso si leggerà invece uno 0.

movimento ORARIO		movimento ANTIORARIO	
posizione di riposo	IN1 = 1, IN2 = 1	posizione di riposo	IN1 = 1, IN2 = 1

inizio movimento orario (A)	IN1 = 0, IN2 = 1	inizio movimento orario (B)	IN1 = 1, IN2 = 0
metà movimento (A+B)	IN1 = 0, IN2 = 0	metà movimento (A+B)	IN1 = 0, IN2 = 0
fine movimento orario (B)	IN1 = 1, IN2 = 0	fine movimento orario (A)	IN1 = 0, IN2 = 1

La tabella del Nutchip collegherà uno stato all'altro tenendo traccia di queste fasi. La chiusura di entrambi i contatti indicherà il momento di partire: ci porteremo in uno stato intermedio, attendendo che la riapertura di uno dei due ci indichi il verso di rotazione. Potremo quindi portarci definitivamente nello stato successivo, dove potremo eventualmente attendere che si riaprano entrambi i contatti (se temiamo che il nostro encoder abbia dei rimbalzi).

LED rotanti

Il primo esperimento è quello di realizzare delle semplici luci rotanti. Girando l'encoder, il LED acceso si muoverà da destra a sinistra o da sinistra a destra. Questo tipo di funzionamento è analogo a quello di un SELETTORE a quattro vie. Quando si avvanza oltre l'ultimo LED si riaccende il primo, e viceversa. Il file da caricare è *encoder_rot.nut*.

LED a barra

In questo esperimento i LED si accendono progressivamente. Questo tipo di visualizzazione dà l'impressione di una grandezza che aumenta o diminuisce ed è adatto per esempio ad un controllo di volume o di velocità di un motore.

A differenza del controllo precedente, quando si avvanza dopo che anche l'ultimo LED si è acceso, la sequenza non ricomincia da capo ma resta come congelata. Il file da caricare è *encoder_bar.nut*

LED a sequenza predefinita

Negli esperimenti precedenti abbiamo disposto le uscite in modo da dare un'accensione "ordinata" ai LED, uno dopo l'altro. In realtà possiamo scrivere nelle uscite una qualsiasi altra sequenza: ad esempio, se usiamo dei LED rossi, gialli e verdi (ed eventualmente un LED tverde a forma di freccia) possiamo simulare la sequenza di un semaforo (solo rosso; verde+freccia; verde+giallo+freccia; rosso+freccia...).

Contatore binario

Un caso speciale di uscite in sequenza predefinita è dato dal contatore binario. Per un contatore binario, le uscite si attivano in una sequenza che rispecchia quella delle cifre dei numeri binari. Una tale sequenza si può usare per pilotare ad esempio un display.

I più esperti avranno già capito come farlo... Potete vederlo nel prossimo progetto, un [display regolabile ad encoder](#).

Azioniamo un motore passo-passo

[Le basi](#) | [Motori passo passo](#) | [Stadi di pilotaggio](#) | [Schema elettrico](#) | [Tavola degli stati](#) | [Half stepping](#) | [Tavola half stepping](#)

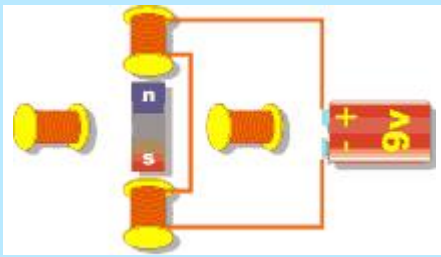
Le basi

Se vogliamo azionare un motore passo passo dobbiamo **prima capire** come funziona: altrimenti non sapremo superare le prime difficoltà. Dovremo poi attrezzarci con uno speciale **stadio di potenza** capace di fornire tutta la corrente assorbita dal motore.

I motori passo passo (*stepper motor*) funzionano facendo ruotare un magnete, attirandolo da un passo all'altro con delle elettrocalamite opportunamente disposte. A differenza degli altri tipi di motore, in uno *stepper* non ci sono contatti striscianti: per ottenere il movimento dobbiamo alternare **dall'esterno** il passaggio della corrente nei suoi due avvolgimenti. Nel farlo, dobbiamo rispettare una **sequenza** ben precisa, composta da 4 passi. Se la sequenza **non** è quella giusta, il motore vibrerà e ronzerà, ma **non girerà**.

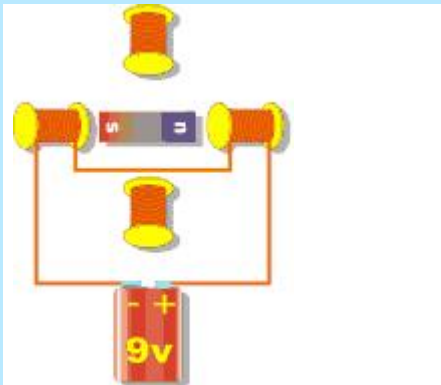
Anche se non sono insoliti motori da **100 o 200 passi per giro**, è sempre possibile ricondursi al prossimo esempio che fa vedere un motore da **4 passi/giro**. Il motore è costruito con 4 elettromagneti, disposti a croce: al centro una calamita è libera di **ruotare**.

1



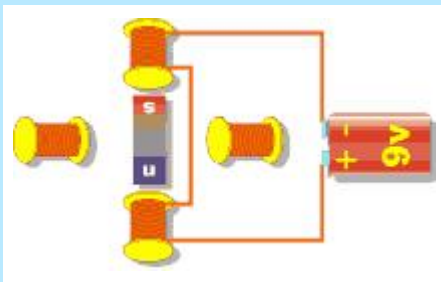
Nel primo passo alimentiamo i due elettromagneti in verticale. I due magneti sono in serie, il solenoide in alto affaccia il polo S alla calamita mentre il solenoide in basso mostra il proprio polo N. Poli opposti si attraggono, perciò la calamita si porterà con il **polo N in alto ed il S in basso**.

2



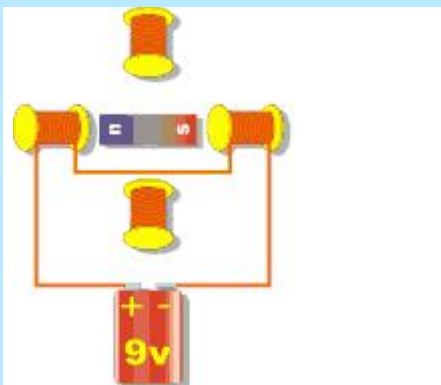
Al passo successivo, alimentiamo i due magneti posti in orizzontale. Anche questi solenoidi sono collegati in serie. La calamita "vedrà" il polo S a destra ed il N a sinistra, e si porterà in questa posizione compiendo un **quarto di giro** in senso orario. Non ruota dalla parte opposta perchè poli uguali si respingono.

3



Al terzo passo colleghiamo ancora l'avvolgimento verticale, ma **invertendo** la polarità della batteria: la corrente circola in senso inverso, il polo S è ora in basso e quello N in alto. La calamita ruota di un altro **quarto di giro** in senso orario, attirata dai poli opposti. La situazione è simmetrica rispetto al primo passo.

4



Nell'ultimo passo si alimentano di nuovo i solenoidi in orizzontale, con polarità **rovesciata** rispetto al secondo passo. La calamita ruota di un ulteriore **quarto di giro**, per avvicinare il proprio polo N al polo S che ora si trova a sinistra, ed il proprio polo S al polo N che ora è a destra.

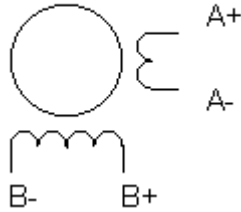
La sequenza prosegue ricominciando dal passo 1: così facendo la calamita tornerà alla posizione iniziale e **avrà compiuto un giro intero**.

Per variare la **velocità** di uno stepper basta variare il **tempo** in cui si rimane nel passo. Per **fermarsi**, invece, ci sono **due** modi:

-) si può lasciare l'alimentazione inserita, in tal caso il motore si bloccherà saldamente nella posizione assunta
-) si può scollegare l'alimentazione, in tal caso il motore sarà libero di girare.

Motori passo passo

MOTORE PASSO-PASSO
STEPPER MOTOR

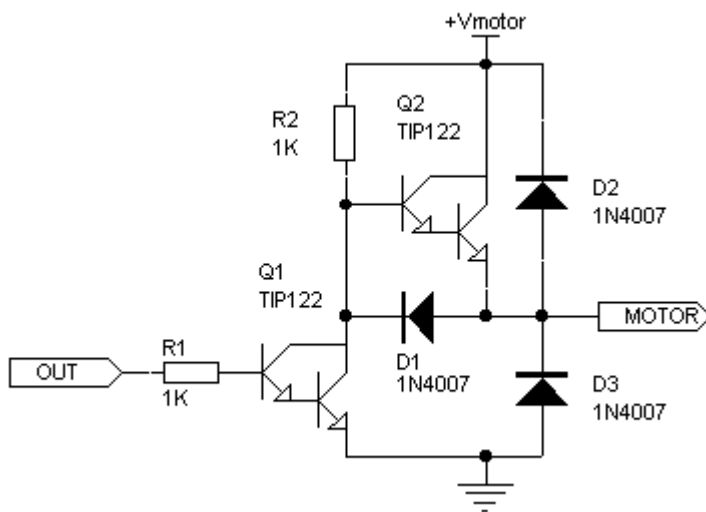


Il simbolo del motore passo passo ricorda l'esempio: i terminali A+A- alimentano i solenoidi verticali, mentre B+B- alimentano quelli orizzontali. Questi motori si chiamano *bipolari*, perchè per pilotarli bisogna invertire la polarità delle tensioni ai loro capi.

-) I motori bipolari si riconoscono perchè hanno 4 fili.
-) Con un tester possiamo separare l'avvolgimento A da quello B verificando la continuità elettrica.
-) Non è importante sapere quale filo sia il + e quale il -, invertendoli il motore funziona ugualmente anche se gira al contrario.

Driver per motori passo passo

Per pilotare un motore passo passo dobbiamo utilizzare uno stadio di potenza (*driver*) capace di erogare la corrente richiesta dagli avvolgimenti. Serve **uno stadio per ogni filo** del motore (4 stadi in tutto). **Ogni** stadio è pilotato da un'**uscita** del NUTCHIP.



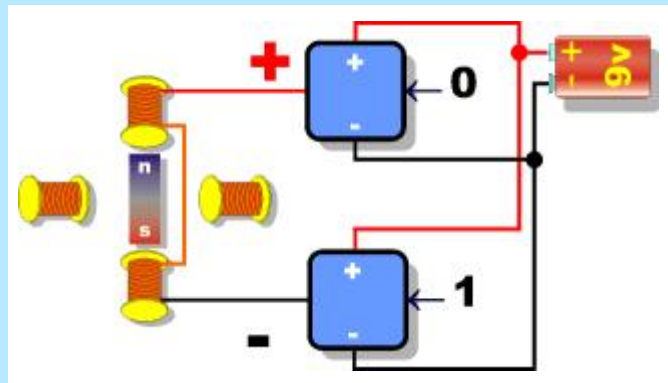
Uno stadio *driver* è composto da **due** transistor darlington Q1 e Q2. Il simbolo del darlington è quello di una *coppia* di transistor.

-) se l'uscita del NUTCHIP (OUT) vale 0, il driver collega il filo del motore al polo positivo
-) se l'uscita del NUTCHIP (OUT) vale 1, il driver collega il filo del motore al polo negativo (GND)

" +Vmotor" è la tensione di alimentazione positiva del motore (tipicamente 5, 9 o 12 volt), indipendente dai +5V che alimentano il NUTCHIP. La massa è in parallelo con la GND del NUTCHIP.

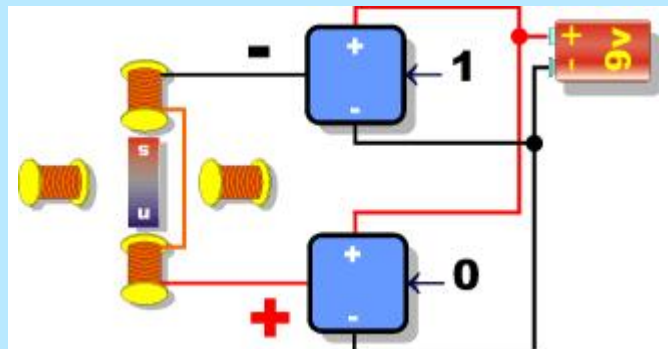
Vediamo cosa succede quando utilizziamo due driver per pilotare l'avvolgimento verticale. Per portarci nel primo passo della sequenza il NUTCHIP dà:

-)] uno 0 al driver in alto, che collega il **polo positivo** al motore e
-)] un 1 al driver in basso, che collega il **polo negativo** al motore.



Per invertire la polarità al motore, come richiesto dal terzo passo della sequenza, il NUTCHIP darà:

-)] un 1 al driver in alto, che collega il **polo negativo** al motore e
-)] uno 0 al driver in basso, che collega il **polo positivo** al motore.

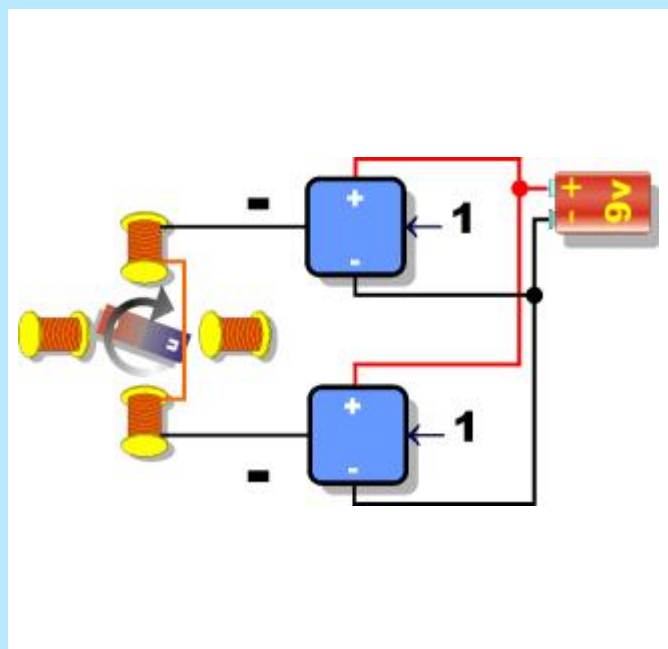


Per lasciare i solenoidi verticali inattivi (come nel passo 2 e 4), il NUTCHIP fornisce ai driver una **coppia di 1**.

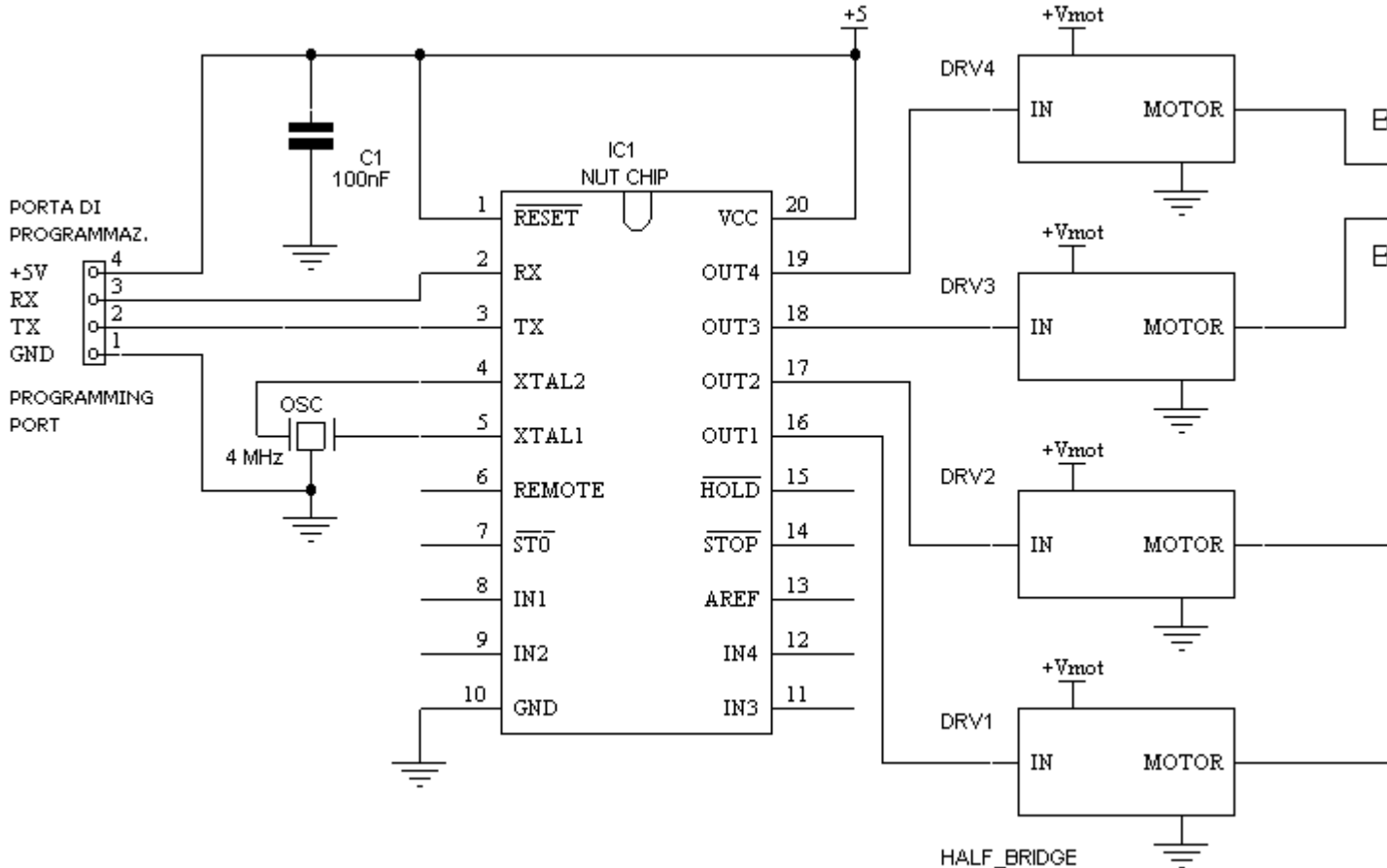
-)] sia il driver in alto che quello in basso collegano il motore al **polo negativo**. Siccome entrambi i fili sono allo **stesso potenziale**, la corrente non circola ed il magnete è libero di muoversi.

Si ottiene lo **stesso risultato** anche pilotando i driver con una **coppia di 0**.

In tal caso, entrambi i fili del motore si portano al **polo positivo**, ma essendo ancora entrambi allo **stesso potenziale**, la corrente non circola lo stesso.



Schema elettrico



Per semplicità, in questo schema ogni stadio driver è disegnato come un blocchetto. Se il vostro motore è a 5 volt, potete collegare la +Vmot alla +5V senza problemi. Altrimenti vi serve un secondo alimentatore per fornire la tensione positiva al motore (il polo positivo va collegato al negativo che alimenta il NUT).

Serve una buona dose di pazienza per montare tutti gli 8 transistor darlington e i 12 diodi necessari. Inoltre bisogna scoprire quali sono i fili A e B del motore passo passo.

-)] Se il vostro motore ha **4 fili**, identificate (con un ohmetro) le due coppie collegate fra loro.
-)] Se ha **5 fili non potete** usarlo con questo circuito.
-)] Se ha **6 fili** è probabile che sia un motore con *prese centrali* (come quello disegnato nello schema). Lasciate le prese centrali *scollegate*. La presa centrale è quella che, con l'ohmetro, dà metà resistenza rispetto agli altri fili collegati.

Potete limitarvi a separare i fili in *due coppie* usando l'ohmetro. Il circuito infatti funziona ugualmente:

-)] scambiando entrambi gli A con entrambi i B
-)] invertendo A+ con A-
-)] invertendo B+ con B-

In tutti questi casi il motore girerà ugualmente, anche se al contrario. Se invece commettete altri errori (ad esempio, scambiate A+ con B+) il motore non girerà.

Tavola degli stati, pilotaggio a 4 passi

Se la spiegazione del circuito è stata lunga, la tavola degli stati che aziona un motore passo passo è sorprendentemente semplice: basta generare quattro stati, uno per ogni passo della sequenza illustrata sopra. Il file relativo è *stepper.nut*. Lo stato 1 genera il passo 1, lo stato 2 il passo 2 e così via. Il timeout fa cambiare automaticamente stato, saltando al passo successivo. Il tempo che si imposta come timeout determina la velocità del motore.

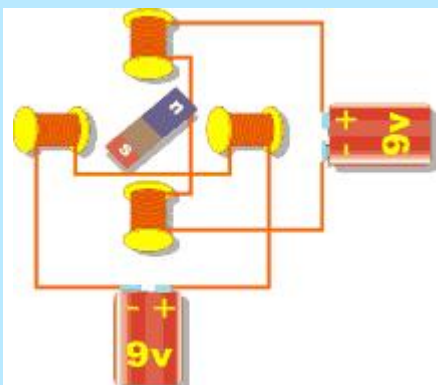
state	output 1...4	input 1...4	remote	next
st00	1 0 0 0	timeout 20 mS		st01
st01	0 0 1 0	timeout 20 mS		st02
st02	0 1 0 0	timeout 20 mS		st03
st03	0 0 0 1	timeout 20 mS		st00

Tavola della verità per azionare uno stepper motor

Mezzi passi

C'è un'altra sequenza idonea a fare ruotare un motore passo passo: la cosiddetta *half step* (mezzo passo). Si ottiene dalla sequenza precedente, in cui fra ogni passo ed il successivo si **aggiunge** un passo creato unendo gli altri due.

Ad esempio, fra il **primo** ed il **secondo** passo si inserisce:



Passo 1 e mezzo:

In questo passo si collegano gli avvolgimenti **sia** del primo passo, **sia** del secondo.
La calamita ruota di **un ottavo di giro**, per avvicinare portare il proprio polo N in una posizione intermedia fra i due poli S che ora si trovano in sia alto e sia a destra.

La sequenza half step è quindi composta da **8 passi** anziché 4. Ogni passo aggiunto è la "somma" del passo precedente con quello seguente.

Tavola degli stati, pilotaggio "half step"

Il file *stepper_half.nut* pilota il motore con la tecnica del mezzo passo.

L'*half stepping* ha vantaggi e svantaggi:

-) **dà molta forza** alla rotazione (per metà del tempo si attivano 2 avvolgimenti anzichè 1)
-) attenzione però a non **surriscaldare** il motore (che deve dissipare l'energia di 2 avvolgimenti al posto di uno solo)
-) la **velocità massima si dimezza** (è raddoppiato il numero di passi)
-) la **precisione di movimento raddoppia** (facciamo due passi dove altrimenti se ne faceva uno).

state	output 1...4	input 1...4	remote	next
st00	1 0 0 0	timeout 20 mS		st01
st01	1 0 1 0	timeout 20 mS		st02
st02	0 0 1 0	timeout 20 mS		st03
st03	0 1 1 0	timeout 20 mS		st04
st04	0 1 0 0	timeout 20 mS		st05
st05	0 1 0 1	timeout 20 mS		st06
st06	0 0 0 1	timeout 20 mS		st07
st07	1 0 0 1	timeout 20 mS		st00

Tavola della verità per azionare uno stepper motor con la tecnica "half step"

Azioniamo un motore passo-passo

[Le basi](#) | [Motori passo passo](#) | [Stadi di pilotaggio](#) | [Schema elettrico](#) | [Tavola degli stati](#) | [Half stepping](#) | [Tavola half stepping](#)

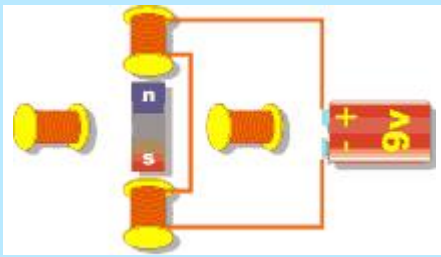
Le basi

Se vogliamo azionare un motore passo passo dobbiamo **prima capire** come funziona: altrimenti non sapremo superare le prime difficoltà. Dovremo poi attrezzarci con uno speciale **stadio di potenza** capace di fornire tutta la corrente assorbita dal motore.

I motori passo passo (*stepper motor*) funzionano facendo ruotare un magnete, attirandolo da un passo all'altro con delle elettrocalamite opportunamente disposte. A differenza degli altri tipi di motore, in uno *stepper* non ci sono contatti striscianti: per ottenere il movimento dobbiamo alternare **dall'esterno** il passaggio della corrente nei suoi due avvolgimenti. Nel farlo, dobbiamo rispettare una **sequenza** ben precisa, composta da 4 passi. Se la sequenza **non** è quella giusta, il motore vibrerà e ronzerà, ma **non girerà**.

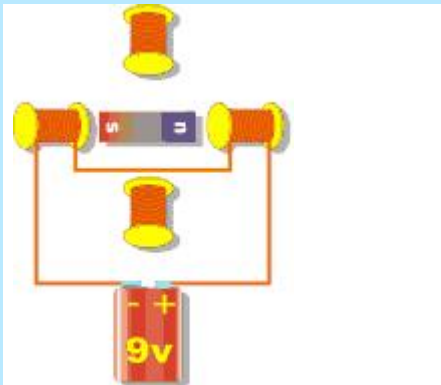
Anche se non sono insoliti motori da **100 o 200 passi per giro**, è sempre possibile ricondursi al prossimo esempio che fa vedere un motore da **4 passi/giro**. Il motore è costruito con 4 elettromagneti, disposti a croce: al centro una calamita è libera di **ruotare**.

1



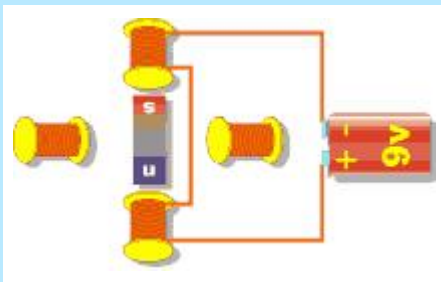
Nel primo passo alimentiamo i due elettromagneti in verticale. I due magneti sono in serie, il solenoide in alto affaccia il polo S alla calamita mentre il solenoide in basso mostra il proprio polo N. Poli opposti si attraggono, perciò la calamita si porterà con il **polo N in alto ed il S in basso**.

2



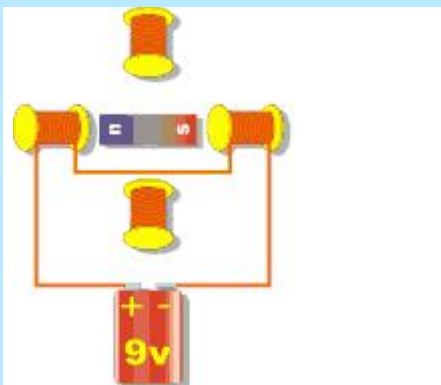
Al passo successivo, alimentiamo i due magneti posti in orizzontale. Anche questi solenoidi sono collegati in serie. La calamita "vedrà" il polo S a destra ed il N a sinistra, e si porterà in questa posizione compiendo un **quarto di giro** in senso orario. Non ruota dalla parte opposta perchè poli uguali si respingono.

3



Al terzo passo colleghiamo ancora l'avvolgimento verticale, ma **invertendo** la polarità della batteria: la corrente circola in senso inverso, il polo S è ora in basso e quello N in alto. La calamita ruota di un altro **quarto di giro** in senso orario, attirata dai poli opposti. La situazione è simmetrica rispetto al primo passo.

4



Nell'ultimo passo si alimentano di nuovo i solenoidi in orizzontale, con polarità **rovesciata** rispetto al secondo passo. La calamita ruota di un ulteriore **quarto di giro**, per avvicinare il proprio polo N al polo S che ora si trova a sinistra, ed il proprio polo S al polo N che ora è a destra.

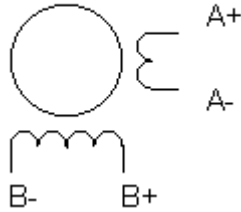
La sequenza prosegue ricominciando dal passo 1: così facendo la calamita tornerà alla posizione iniziale e **avrà compiuto un giro intero**.

Per variare la **velocità** di uno stepper basta variare il **tempo** in cui si rimane nel passo. Per **fermarsi**, invece, ci sono **due** modi:

-)] si può lasciare l'alimentazione inserita, in tal caso il motore si bloccherà saldamente nella posizione assunta
-)] si può scollegare l'alimentazione, in tal caso il motore sarà libero di girare.

Motori passo passo

MOTORE PASSO-PASSO
STEPPER MOTOR

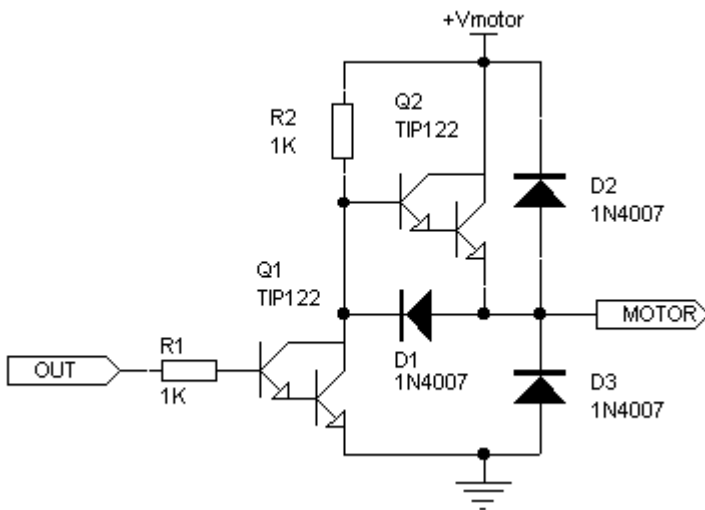


Il simbolo del motore passo passo ricorda l'esempio: i terminali A+A- alimentano i solenoidi verticali, mentre B+B- alimentano quelli orizzontali. Questi motori si chiamano *bipolari*, perchè per pilotarli bisogna invertire la polarità delle tensioni ai loro capi.

-) I motori bipolari si riconoscono perchè hanno 4 fili.
-) Con un tester possiamo separare l'avvolgimento A da quello B verificando la continuità elettrica.
-) Non è importante sapere quale filo sia il + e quale il -, invertendoli il motore funziona ugualmente anche se gira al contrario.

Driver per motori passo passo

Per pilotare un motore passo passo dobbiamo utilizzare uno stadio di potenza (*driver*) capace di erogare la corrente richiesta dagli avvolgimenti. Serve **uno stadio per ogni filo** del motore (4 stadi in tutto). **Ogni** stadio è pilotato da un'**uscita** del NUTCHIP.



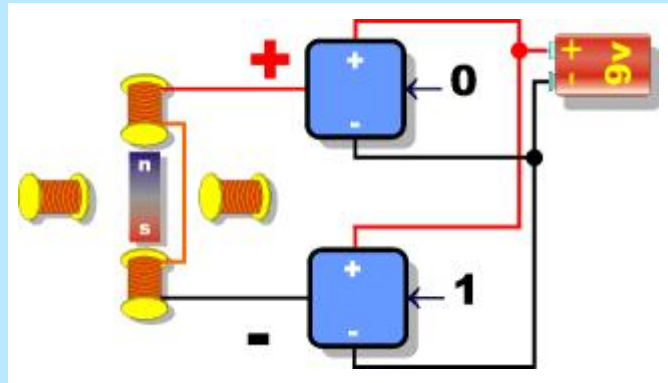
Uno stadio *driver* è composto da **due** transistor darlington Q1 e Q2. Il simbolo del darlington è quello di una *coppia* di transistor.

-) se l'uscita del NUTCHIP (OUT) vale 0, il driver collega il filo del motore al polo positivo
-) se l'uscita del NUTCHIP (OUT) vale 1, il driver collega il filo del motore al polo negativo (GND)

" +Vmotor" è la tensione di alimentazione positiva del motore (tipicamente 5, 9 o 12 volt), indipendente dai +5V che alimentano il NUTCHIP. La massa è in parallelo con la GND del NUTCHIP.

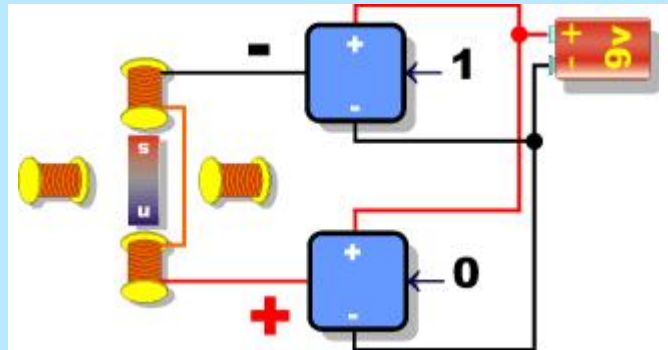
Vediamo cosa succede quando utilizziamo due driver per pilotare l'avvolgimento verticale. Per portarci nel primo passo della sequenza il NUTCHIP dà:

-)] uno 0 al driver in alto, che collega il **polo positivo** al motore e
-)] un 1 al driver in basso, che collega il **polo negativo** al motore.



Per invertire la polarità al motore, come richiesto dal terzo passo della sequenza, il NUTCHIP darà:

-)] un 1 al driver in alto, che collega il **polo negativo** al motore e
-)] uno 0 al driver in basso, che collega il **polo positivo** al motore.

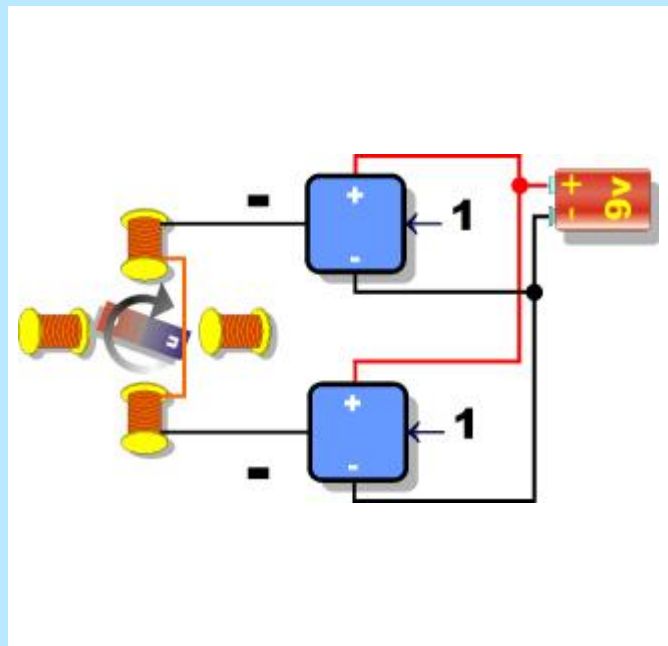


Per lasciare i solenoidi verticali inattivi (come nel passo 2 e 4), il NUTCHIP fornisce ai driver una **coppia di 1**.

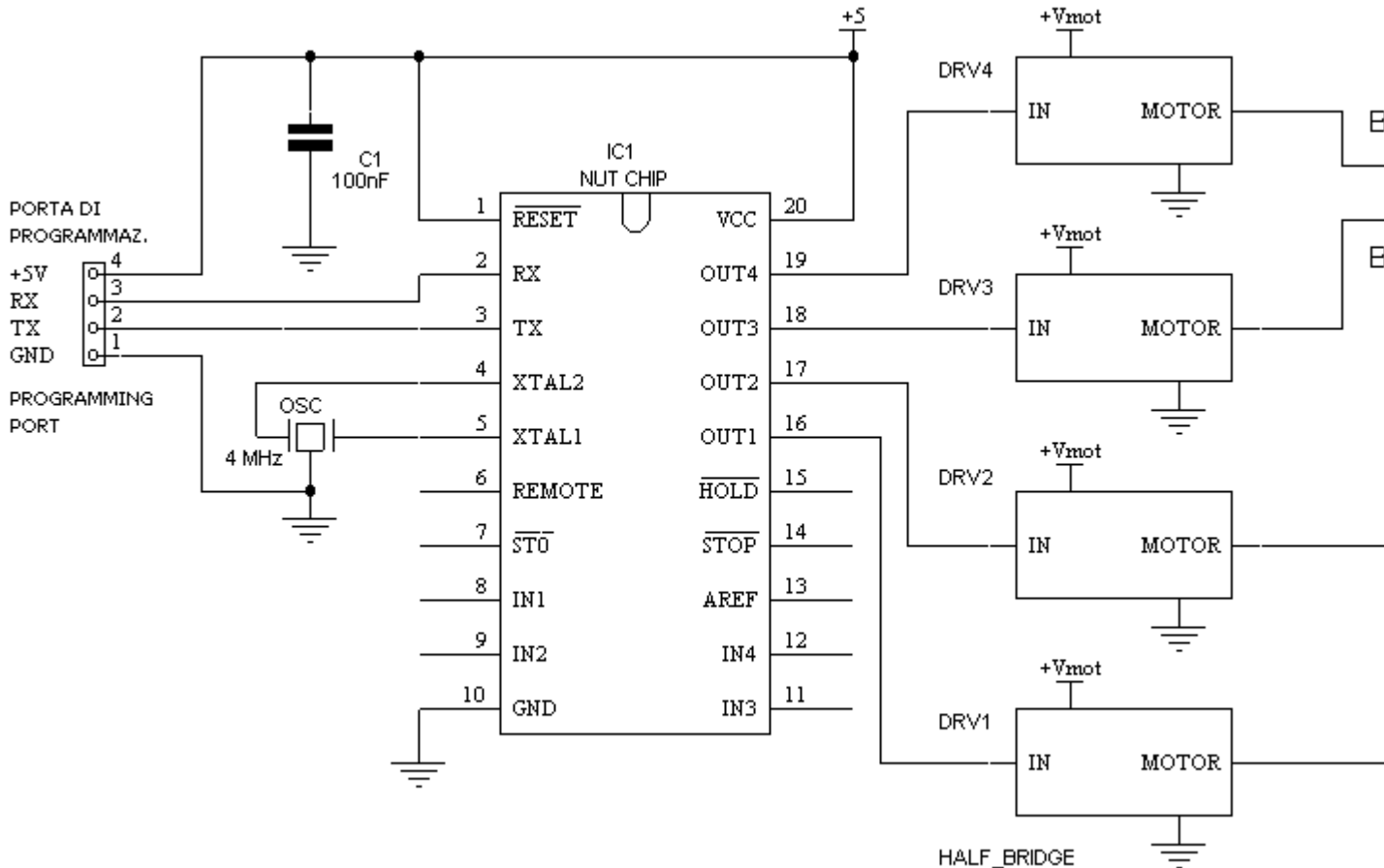
-)] sia il driver in alto che quello in basso collegano il motore al **polo negativo**. Siccome entrambi i fili sono allo **stesso potenziale**, la corrente non circola ed il magnete è libero di muoversi.

Si ottiene lo **stesso risultato** anche pilotando i driver con una **coppia di 0**.

In tal caso, entrambi i fili del motore si portano al **polo positivo**, ma essendo ancora entrambi allo **stesso potenziale**, la corrente non circola lo stesso.



Schema elettrico



Per semplicità, in questo schema ogni stadio driver è disegnato come un blocchetto. Se il vostro motore è a 5 volt, potete collegare la +Vmot alla +5V senza problemi. Altrimenti vi serve un secondo alimentatore per fornire la tensione positiva al motore (il polo positivo va collegato al negativo che alimenta il NUT).

Serve una buona dose di pazienza per montare tutti gli 8 transistor darlington e i 12 diodi necessari. Inoltre bisogna scoprire quali sono i fili A e B del motore passo passo.

-)] Se il vostro motore ha **4 fili**, identificate (con un ohmetro) le due coppie collegate fra loro.
-)] Se ha **5 fili non potete** usarlo con questo circuito.
-)] Se ha **6 fili** è probabile che sia un motore con *prese centrali* (come quello disegnato nello schema). Lasciate le prese centrali *scollegate*. La presa centrale è quella che, con l'ohmetro, dà metà resistenza rispetto agli altri fili collegati.

Potete limitarvi a separare i fili in *due coppie* usando l'ohmetro. Il circuito infatti funziona ugualmente:

-)] scambiando entrambi gli A con entrambi i B
-)] invertendo A+ con A-
-)] invertendo B+ con B-

In tutti questi casi il motore girerà ugualmente, anche se al contrario. Se invece commettete altri errori (ad esempio, scambiate A+ con B+) il motore non girerà.

Tavola degli stati, pilotaggio a 4 passi

Se la spiegazione del circuito è stata lunga, la tavola degli stati che aziona un motore passo passo è sorprendentemente semplice: basta generare quattro stati, uno per ogni passo della sequenza illustrata sopra. Il file relativo è *stepper.nut*. Lo stato 1 genera il passo 1, lo stato 2 il passo 2 e così via. Il timeout fa cambiare automaticamente stato, saltando al passo successivo. Il tempo che si imposta come timeout determina la velocità del motore.

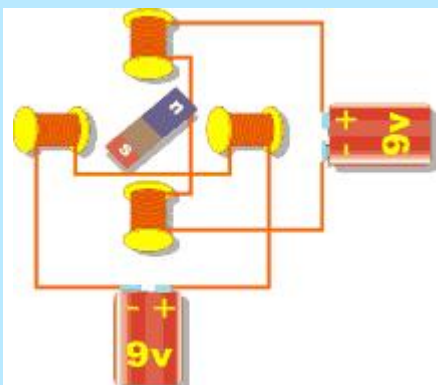
state	output 1...4	input 1...4	remote	next
st00	1 0 0 0	timeout 20 mS		st01
st01	0 0 1 0	timeout 20 mS		st02
st02	0 1 0 0	timeout 20 mS		st03
st03	0 0 0 1	timeout 20 mS		st00

Tavola della verità per azionare uno stepper motor

Mezzi passi

C'è un'altra sequenza idonea a fare ruotare un motore passo passo: la cosiddetta *half step* (mezzo passo). Si ottiene dalla sequenza precedente, in cui fra ogni passo ed il successivo si **aggiunge** un passo creato unendo gli altri due.

Ad esempio, fra il **primo** ed il **secondo** passo si inserisce:



1
-
2

Passo 1 e mezzo:

In questo passo si collegano gli avvolgimenti **sia** del primo passo, **sia** del secondo.
La calamita ruota di **un ottavo di giro**, per avvicinare portare il proprio polo N in una posizione intermedia fra i due poli S che ora si trovano in sia alto e sia a destra.

La sequenza half step è quindi composta da **8 passi** anziché 4. Ogni passo aggiunto è la "somma" del passo precedente con quello seguente.

Tavola degli stati, pilotaggio "half step"

Il file *stepper_half.nut* pilota il motore con la tecnica del mezzo passo.

L'*half stepping* ha vantaggi e svantaggi:

-) **dà molta forza** alla rotazione (per metà del tempo si attivano 2 avvolgimenti anzichè 1)
-) attenzione però a non **surriscaldare** il motore (che deve dissipare l'energia di 2 avvolgimenti al posto di uno solo)
-) la **velocità massima si dimezza** (è raddoppiato il numero di passi)
-) la **precisione di movimento raddoppia** (facciamo due passi dove altrimenti se ne faceva uno).

state	output 1...4	input 1...4	remote	next
st00	1 0 0 0	timeout 20 mS		st01
st01	1 0 1 0	timeout 20 mS		st02
st02	0 0 1 0	timeout 20 mS		st03
st03	0 1 1 0	timeout 20 mS		st04
st04	0 1 0 0	timeout 20 mS		st05
st05	0 1 0 1	timeout 20 mS		st06
st06	0 0 0 1	timeout 20 mS		st07
st07	1 0 0 1	timeout 20 mS		st00

Tavola della verità per azionare uno stepper motor con la tecnica "half step"