

# NUTCHIP

I **Nutchip** sono integrati **universali**. Sono **facili** da usare in **mille** applicazioni.

Sono **differenti** perchè usano le **tavole della verità** al posto di un linguaggio di programmazione.

Infatti basta riempire la **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.

Potete **riprogrammare** il chip tutte le volte che volete, un solo chip serve per migliaia di esperimenti: una [fotocellula](#), un [timer](#), un [antifurto](#) senza fili, un [telecomando](#), un controllo per [motori passo passo](#), un [dado elettronico](#), un [gioco](#) di riflessi, un termostato... Sono tutti progetti **facili da costruire** con un Nutchip.

## Caratteristiche tecniche

CARATTERISTICHE TECNICHE	
TELECOMANDO	CODIFICA A RAGGI INFRAROSSI OPPURE VIA RADIO FINO 6 PULSANTI
TIMER	DIGITALE DA 1 mS a 1000 h PROGRAMMABILE PASSO PER PASSO FINO A 30 TEMPI DIFFERENTI PER PROGRAMMA
INGRESSI	4 DIGITALI PROGRAMMABILI COLLEGABILI DIRETTAMENTE A PULSANTI O A INTEGRATI LOGICI
USCITE	4 DIGITALI PROGRAMMABILI POTENZA: 5 V, 10 mA cad.
MATRICE DIGITALE PROGRAMMABILE	4 INGRESSI x 4 USCITE: FINO A 64 COMBINAZIONI DI INGRESSI FINO A 32 COMBINAZIONI DI USCITE
COMPARATORE ANALOGICO	A 2 INGRESSI PER TENSIONI DA 0 FINO A 5 V (COLLEGATO ALL'INGRESSO 4)
INGRESSI SUPPLEMENTARI	- MARCIA/ARRESTO DEL TIMER - BLOCCO USCITE - RESET
FREQUENZA	4 MHz CON QUARZO OPPURE OSCILLATORE CERAMICO
ALIMENTAZIONE	5 VOLT (STABILIZZATA)
DIMENSIONI	CIRCUITO INTEGRATO A 20 PIEDINI D.I.L. COMPATIBILE PER BASETTE SPERIMENTALI SENZA SALDATURE
PROGRAMMAZIONE	PORTA SERIALE DEL PC TRAMITE ADATTATORE SISTEMA ICP: NON OCCORRE SMONTARE IL CHIP DAL CIRCUITO

Ogni Nutchip contiene:

- un decodificatore di **telecomandi**
- una **matrice digitale** programmabile a **4 ingressi** e **4 uscite**
- un **timer** regolabile da 1 millesimo di secondo a mille ore
- un **comparatore** capace di valutare un sensore
- 3 ingressi digitali **preprogrammati**.

Gli **ingressi** accettano pulsanti, interruttori, contatti di ogni tipo o circuiti digitali. **Accendete** dei LED collegandoli alle **uscite**, oppure **pilotate** relè, lampade, cicalini, elettrocalamite, motori attraverso semplici circuiti. Fotorisistenze, **sensori** di temperatura, di suoni, di pressione vanno collegati al comparatore analogico.

Il chip funziona sia con **telecomandi** ad **infrarossi** che **via radio**, cambiando semplicemente il ricevitore esterno, che è disponibile come **modulo** preassemblato.

**Non servono linguaggi di programmazione**, per questo sono facili da usare. Basta riempire la **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 del computer. Con la tecnologia ICP (in-circuit programming) la programmazione avviene senza smontare il chip dal circuito finale.



(1) compilate la tabella per configurare la matrice ingressi/uscite



(2) scegliete tipo e pulsanti del telecomando



(3) programmate il chip attraverso la porta seriale del PC

## Sguardo d'insieme

NUTCHIP è il nome di una famiglia di **chip universali** per piccoli automatismi. Un Nutchip e qualche componente esterno è tutto ciò che serve per realizzare migliaia di circuiti diversi, tutti **facili e divertenti**. Un [telecomando](#), un antifurto, una [fotocellula](#), un timer, un [dado elettronico](#), un metronomo, un generatore di impulsi, un termostato, un campanello intelligente: sono solo alcuni esempi di ciò che si può realizzare con i Nutchip.

I Nutchip sono **si utilizzano da subito**. Se siete in grado di leggere una tabella della verità allora siete già in grado di programmare un Nutchip. I Nutchip utilizzano particolari **tabelle della verità** dette *macchine a stati* al posto di un linguaggio di programmazione.

Per **appassionati** e **studenti** i Nutchip sono un'ottima **esperienza per muovere i primi passi** nel mondo della programmazione; mentre gli **esperti li trovano insostituibili** per realizzare in pochi minuti ogni genere di automatismo.

caratteristiche tecniche	pinout del Nutchip
<ul style="list-style-type: none"> <li>▶ <a href="#">nessun linguaggio di programmazione</a>, basta introdurre una <a href="#">tabella della verità</a></li> <li>▶ versatile versatile <a href="#">multi-timer</a> programmabile da 1 millesimo di secondo a ben 1000 ore!</li> <li>▶ decodificatore per <a href="#">telecomando</a> incorporato, fino a 6 tasti programmabili a piacere. Riconosce sia i codici a radiofrequenza sia i codici a infrarossi</li> <li>▶ <a href="#">comparatore analogico</a> incorporato per fissare soglie, collegare direttamente sensori, etc.</li> <li>▶ 3 <a href="#">ingressi digitali</a>, oppure (il comparatore si può utilizzare come quarto ingresso digitale)</li> <li>▶ 4 <a href="#">uscite digitali</a> ad alta corrente capaci di pilotare direttamente i LED (fino a 20 mA per ogni uscita)</li> <li>▶ 3 <a href="#">ingressi dedicati</a> per gli automatismi meccanici: restart, hold, stop</li> <li>▶ porta seriale per <a href="#">collegamento al PC</a></li> <li>▶ programmabile più 10.000 volte in-circuit, cioè con il chip già montato nel circuito di destinazione, senza necessità di smontarlo né cancellarlo</li> <li>▶ versatile base dei tempi: a quarzo per la massima precisione oppure con un economico oscillatore ceramico</li> <li>▶ alimentazione da 4 a 6 volt stabilizzata</li> </ul>	<p style="text-align: center;">cliccare qui per lo <a href="#">schema a blocchi</a> e le caratteristiche elettriche</p>

## Nutstation è il software per programmare i Nutchip [ [help](#) ]



Per scrivere la **tavola della verità** utilizziamo **Nutstation**, un editor specifico che gira sul PC. Le funzioni svolte di volta in volta dal NUTCHIP sono dettate dalla tavola della verità caricata al suo interno. Possiamo aggiungere commenti e spiegazioni ad ogni riga, ed anche inserire uno o più temporizzatori da **1 millesimo di secondo fino a 1000 ore**. Nutstation mostra la tavola sullo schermo, e ci aiuta ad inserire i vari bit ordinatamente, uno dopo l'altro. Nutstation controlla i dati inseriti in **tempo reale**, e sottolinea **automaticamente** le caselle incomplete o errate.



Nutstation gestisce anche i codici del **telecomando** che vogliamo utilizzare. I codici dei telecomandi TV ed RF più diffusi sono già inseriti, ma li possiamo variare a nostro piacimento per adattarli al telecomando che usiamo. E se non conosciamo quale codice trasmette il nostro telecomando? Niente paura: Nutstation dispone anche una funzione di **autoapprendimento** che legge il codice direttamente dal Nutchip.

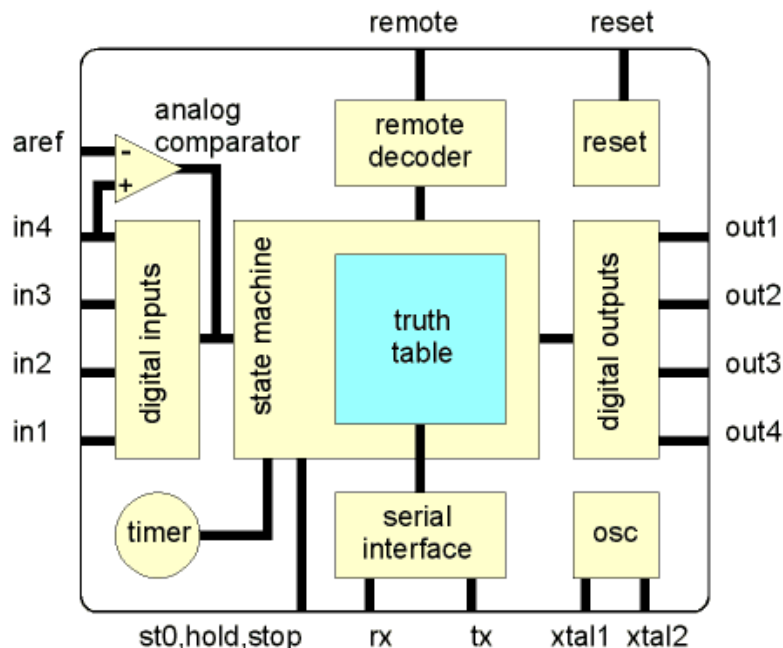


Per programmare un Nutchip non servono apparecchi complicati, ma un **semplice PC con Nutstation**. Un cavo seriale e due transistor bastano per realizzare [l'interfaccia](#) fra il PC ed il chip. Ogni Nutchip si **riprogramma migliaia di volte**: un solo Nutchip perciò si **usa e si riusa** per mille esperienze sempre diverse.

Quando siamo pronti per programmare, colleghiamo il circuito alla porta seriale del PC e premiamo l'apposito pulsante dalla pagina di programmazione di Nutstation.

La programmazione impegna **pochi secondi** e si può fare **anche** con il chip già **saldato nel circuito finale**: sperimentazione, messa a punto, aggiornamenti e modifiche non sono un problema con i Nutchip.

## SCHEMA A BLOCCHI



# Programmazione dei Nutchip.

## L'interfaccia per il PC

I NUTCHIP si programmano serialmente con qualsiasi PC. Tuttavia, poichè i NUTCHIP sono alimentati a 5V mentre la porta seriale dei PC può arrivare fino a  $\pm 12V$ , è necessaria un'interfaccia per potere adattare i diversi livelli di tensione. Per realizzare l'interfaccia bastano due transistor, ma vi fornisco anche degli schemi alternativi nel caso preferiate usare un chip specifico come il MAX232 oppure delle comuni porte logiche.

## Niente porta seriale? Usa l'USB!

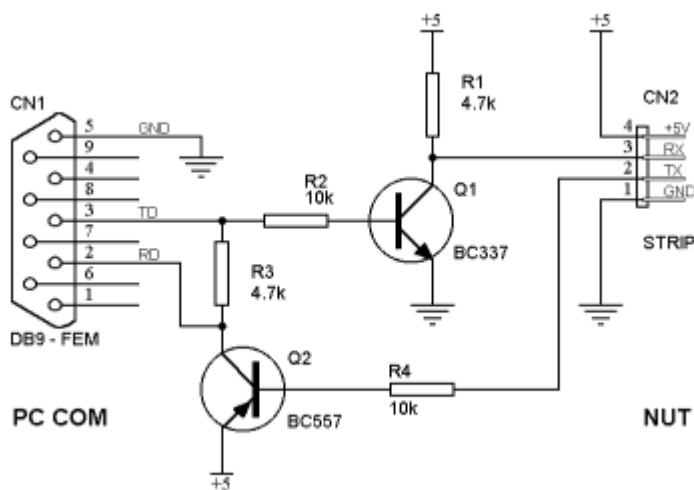
Alcuni PC, specialmente i portatili, non hanno la porta seriale. In questo caso si può usare un convertitore da USB a seriale. Ma attenzione, alcuni dei dispositivi attualmente in commercio non offrono una implementazione completa della porta, altri sono progettati solo per modem o tavolette grafiche, altri ancora per collegare cellulari o macchine fotografiche... Perciò il miglior consiglio è quello di chiedere al negoziante di provare prima dell'acquisto, ma non è facile trovare commercianti disponibili! Per questo motivo abbiamo provato per voi alcuni convertitori commerciali, ed abbiamo raccolto i risultati in questa pagina. [Clicca qui](#) per vedere i risultati!



## Interfaccia consigliata: solo 2 transistor

E' l'interfaccia più **economica** e **semplice**. Utilizza componenti **reperibili** ovunque.

**Abbiamo usato questa interfaccia** per realizzare **tutti** i progetti contenuti in questa documentazione.



*Interfaccia per programmare il Nutchip attraverso la porta seriale del computer.*

### ELENCO COMPONENTI

- Q1 = transistor BC337
- Q2 = transistor BC557
- R1, R3= resistenza 4,7 kohm
- R2, R4= resistenza 10 kohm
- 1 strip di contatti a 4 poli oppure 4 spezzoncini di filo rigido
- 1 connettore a vaschetta da 9 poli, femmina, tipo seriale (chiamato anche "Canon" oppure "D")
- 1 pezzetto di basetta sperimentale tipo "millefori"

### PER GLI ESPERTI E CURIOSONI

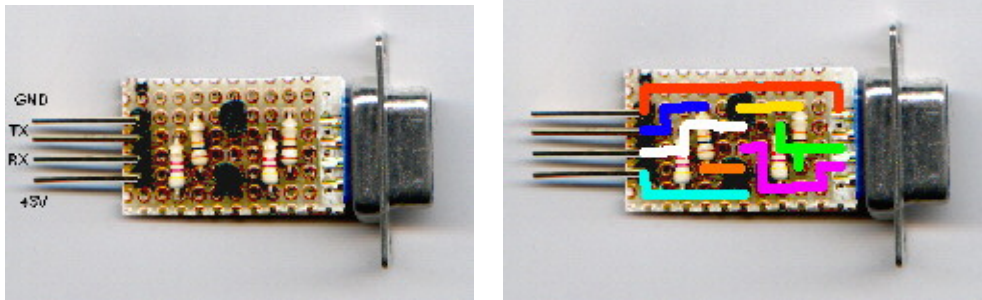
Ecco i risultati della simulazione:

- [trasmissione da PC a Nutchip](#)
- [trasmissione da Nutchip a PC](#)
- [linea a riposo](#)

Questo circuito usa soltanto 2 transistor e sfrutta un trucco per ottenere una tensione negativa. Notate che l'emettitore di Q2 è collegato al positivo, perchè si tratta di un transistor PNP, mentre quello di Q1 è al negativo, perchè si tratta di un transistor NPN.

Quando il PC non trasmette fornisce una tensione **negativa** sul piedino TX: siccome con i Nutchip la **trasmissione** e la **ricezione** non si sovrappongono mai, tramite la resistenza R3 questa tensione negativa **torna** al PC sul piedino RD (ricezione). Quando il NUTCHIP ha un dato da trasmettere, il transistor Q2 commuta fra la tensione **negativa** così "rubata" e la tensione positiva di **alimentazione** di 5V.

Nel 99% dei PC questo segnale, seppure non standard, risulta perfettamente adeguato. Un bel risparmio di soldi e di spazio!

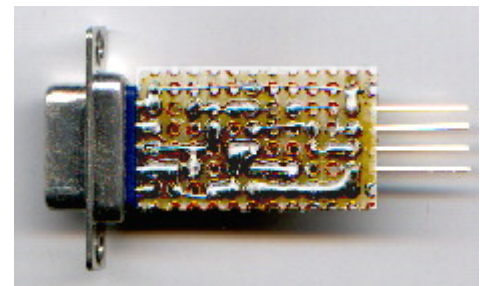


*Realizzazione dell'interfaccia su una basetta sperimentale (a) con le connessioni saldate poste in evidenza (b)*

*Notate che il connettore è saldato solo sotto: dal lato componenti il connettore a 9 poli **non** ha saldature, e le piazzole sono limate via per evitare corto circuiti.*

L'interfaccia finita si innesta direttamente nel cavo seriale; il connettore tipo strip si innesta direttamente nel circuito dove è montato il Nutchip. Utilizzate una strip **lunga** come in figura, in modo da passare **sopra** ai componenti eventualmente presenti sul prototipo da programmare.

Questa interfaccia si inserisce facilmente anche su una basetta a molla per esperimenti ([esempi](#)), e persino su uno zoccolo per integrati da 4+4 piedini tagliato a metà.



*realizzazione delle saldature*

<p>Q1 = BC337 (NPN)</p>	<p>Q2 = BC557 (PNP)</p>	<p><i>Numeri dei piedini sul connettore DB9 femmina. Attenzione perchè il maschio ha una numerazione speculare.</i></p>
-------------------------	-------------------------	---

## Connettori da usare

- Se l'interfaccia è **incorporata** nel circuito, usate un connettore standard per porte seriali (tipo "D" femmina a 9 pin). Una comune prolunga maschio-femmina sarà tutto ciò che serve per collegarsi ad un PC e non servono adattatori speciali.
- Se l'interfaccia è **esterna**, usate un connettore che possa accogliere una "strip" a 4 pin, collegato come segue:
  - 1 = GND
  - 2 = TX
  - 3 = RX
  - 4 = +5V

Se userete una di queste due connessioni non avrete problemi, perchè questi connettori sono stati scelti per la **massima economia e praticità**. Utilizzandoli non dovrete mai usare adattatori o cercare quel foglietto di appunti che, al momento buono, non si trova mai. Potrete inoltre **scambiare** un prototipo con i vostri amici, **riprogrammare** un Nutchip con qualsiasi computer o utilizzare **kit** commerciali senza perdite di tempo e senza grovigli di cavi.

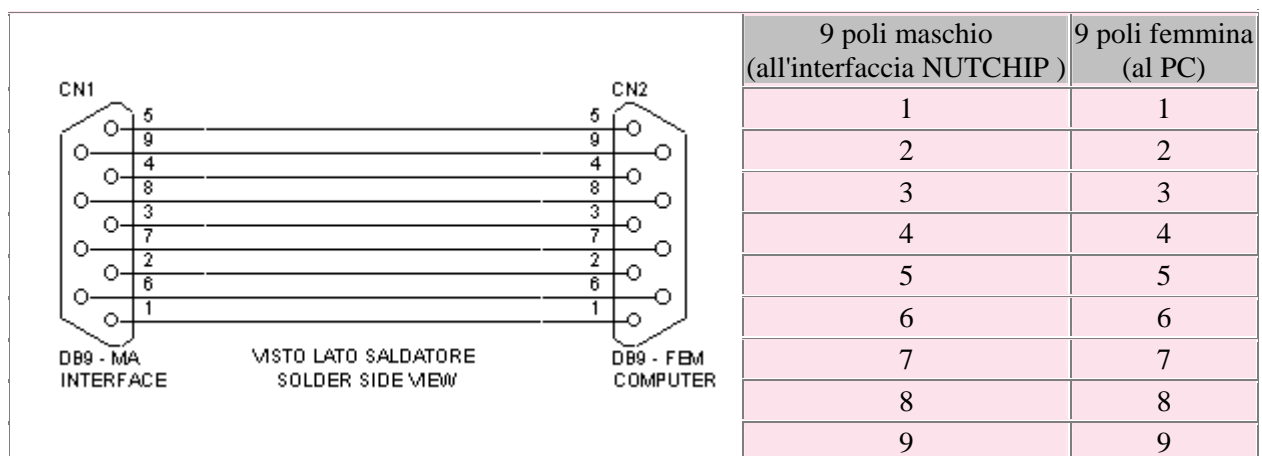
Abbiamo scelto la connessione tipo "strip" perchè:

<ul style="list-style-type: none"> <li>• può essere realizzata da tutti tagliando a metà uno zoccolo per integrati a 4+4 piedini.</li> </ul>	<ul style="list-style-type: none"> <li>• è compatibile con le basette sperimentali a molla (springboard), molto comode per fare esperimenti</li> </ul>
<ul style="list-style-type: none"> <li>• è compatibile con le apposite strip tornite "a petalo", più lunghe, che offrono un contatto eccellente</li> </ul>	<ul style="list-style-type: none"> <li>• per uso professionale, sono disponibili versioni "boxed", non invertibili</li> </ul>

Però fate **ATTENZIONE** a **non invertire** il connettore! Generalmente un'inversione del connettore di programmazione **non danneggia** il prototipo.

## Cavo seriale

Per programmare i NUTCHIP usate un **comune cavo di prolunga** seriale reperibile ovunque. Ecco come è fatto:



**Non** usate i cavi chiamati "null modem" perchè non funzionano. Se usate l'interfaccia a **transistor** potete limitarvi a saldare i fili 2,3 e 5.

## Corrispondenza dei pin connettore 25 poli - 9 poli

Alcuni PC dispongono di un connettore a 25 poli femmina anzichè a 9 poli maschio per la porta seriale. Se non trovate un adattatore idoneo in commercio, ecco i dettagli per realizzarne uno:

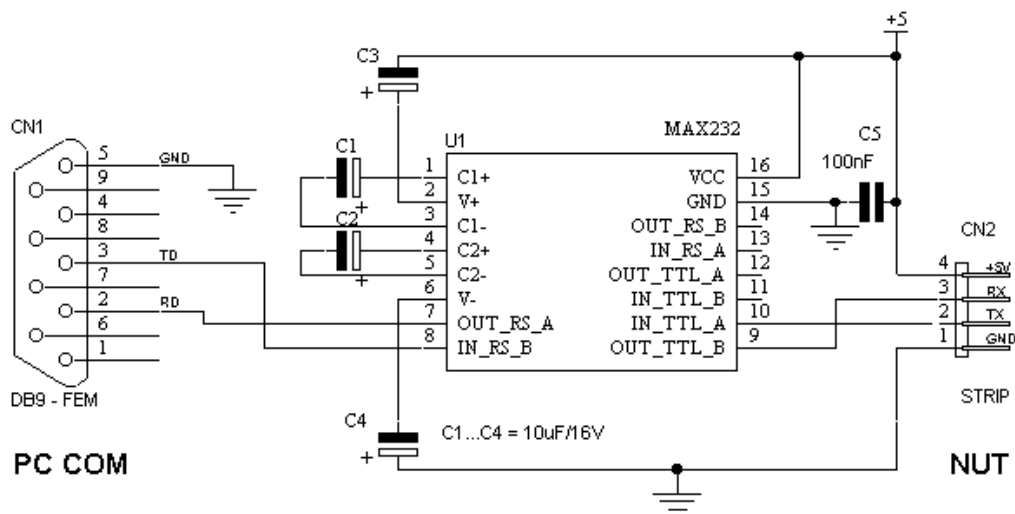
25 poli (femmina sul PC)	9 poli (maschio sul PC)	nome linea RS232	direzione
2	3	TD	usc. PC - ingr. NUTCHIP (RX)
3	2	RD	ingr. PC - usc. NUTCHIP (TX)
4	7	RTS	uscita del PC
5	8	CTS	ingresso del PC
6	6	DSR	ingresso del PC
7	5	GND	MASSA (GND)
8	1	DCD	ingresso del PC
20	4	DTR	uscita del PC
22	9	RI	ingresso del PC

Per innestarsi nel PC, vi serviranno quindi un connettore a 25 poli maschio ed un 9 poli femmina. Se utilizzate l'interfaccia a **transistor** descritta in precedenza, potete limitarvi a saldare solo **tre fili**:

- 1) dal pin 2 del 25 poli al pin 3 del 9 poli;
- 2) dal pin 3 del 25 poli al pin 2 del 9 poli;
- 3) dal pin 7 del 25 poli al pin 5 del 9 poli.

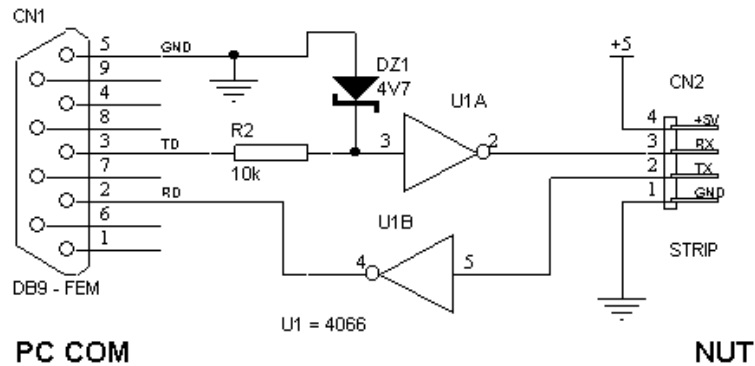
## Interfaccia alternativa: con MAX232

Questa interfaccia utilizza un integrato apposito e fornisce livelli di uscita **standard** adatti per ogni tipo di PC. Trattandosi di un integrato specifico, ha anche la migliore reiezione ai disturbi e la protezione contro scariche e corto circuiti. Per contro è una soluzione più costosa e ingombrante delle soluzioni che utilizzano i transistor o semplici porte logiche.

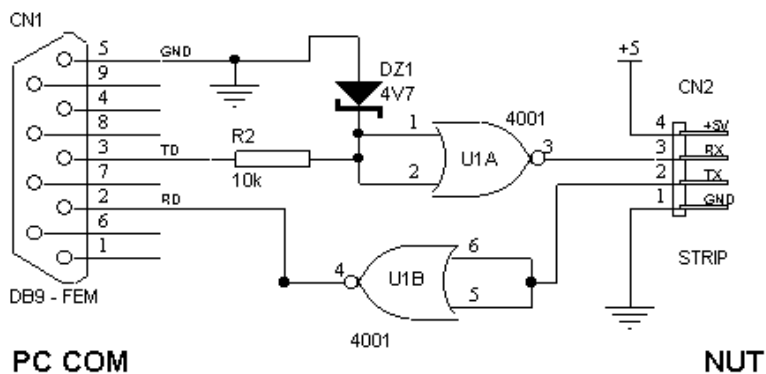


## Interfaccia alternativa: con porte logiche

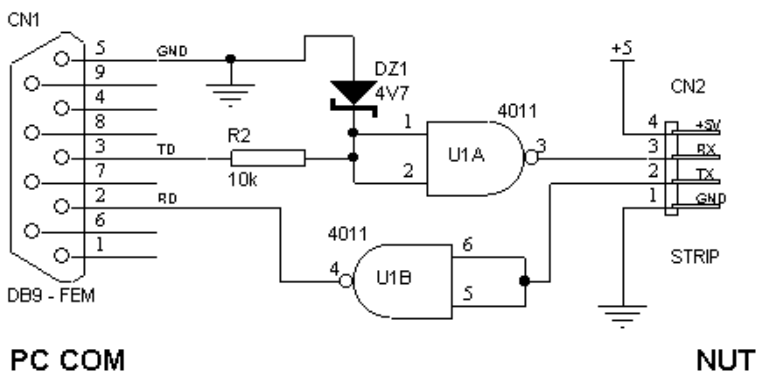
Se nel vostro circuito ci sono alcune porte logiche d'avanzo, potete riutilizzarle come in questo esempio che usa due dei sei inverter di un 4066:



Utilizzando integrati CMOS della serie 4000, scegliete sempre le versioni B (*buffered*) che offrono la maggiore corrente di uscita. Qualsiasi inverter va bene, anche ricavato da porte logiche come NAND e NOR negli esempio che seguono che utilizzano due delle quattro porte di un 4001 o di un 4011:



Non dimenticate di alimentare gli integrati logici collegando gli opportuni piedini al +5V e a GND!



Di tutti i circuiti presentati, questo è il più spartano. Non eroga la tensione negativa richiesta dallo standard, ma soltanto la necessaria **inversione** logica. E' un dato di fatto fatto che la maggioranza dei PC funziona ugualmente bene con un segnale del genere, ma ricordate che si tratta di una piccola forzatura! Perciò **non** è consigliabile usarlo per progetti dove sia richiesta la massima affidabilità e flessibilità, dove i cavi di collegamento siano lunghi e se non conoscete a priori quale PC verrà usato per la programmazione.

## Interfaccia incorporata nella scheda o esterna?

Non c'è una regola valida per tutti.

- Se il vostro circuito deve eseguire una funzione che **non dovrà essere più modificata**, il chip potrà essere programmato **prima** del montaggio sul circuito. Potete quindi semplificare il circuito finale eliminando qualsiasi genere di interfaccia per il PC.
- Se sarete sempre voi a programmare il circuito, allora potete realizzare **una sola interfaccia esterna** da utilizzare per **tutti** i circuiti, collegandola al connettore **STRIP** consigliato (potrebbe bastare anche soltanto metà di uno zoccolo per integrati) che prevederete sulla scheda. Questa è la **soluzione preferita dagli hobbysti** perchè consente di sperimentare nuove soluzioni senza togliere il chip dal circuito.
- Se invece il circuito dovrà essere programmato da altre persone e con un PC che non sia noto a priori, allora è consigliabile montare **tutta l'interfaccia** sulla scheda ed uscire con il connettore a **9 poli standard** per porta seriale. In questo caso, potrete utilizzare qualsiasi PC per riprogrammare il Nutchip: basterà un **comune cavo seriale**, reperibile ovunque. Questa è la soluzione **scelta dai professionisti** quando vogliono che si il loro cliente ad effettuare l'aggiornamento del Nutchip.
- Ovviamente potrete anche decidere di montare il Nutchip su zoccolo e di sfilarlo ogni volta che volete programmarlo: questa soluzione è **obbligatoria** in quei casi dove sul circuito in funzione siano presenti **tensioni pericolose**, come nelle accensioni a triac non isolate.

*una tavola della verità guida il funzionamento del Nutchip*

## La tavola degli stati

### Tavole della verità

Ogni integrato logico, come ad esempio come le porte AND, OR, NAND e NOR, oppure i FLIP-FLOP, funziona obbedendo ad una tavola della verità prestabilita. La tavola della verità riporta, sotto forma di zeri e di uno, tutti i valori che assumeranno le uscite in corrispondenza dei valori presenti agli ingressi.

TAVOLA DEL NAND		
IN1	IN2	OUT
0	0	1
0	1	1
1	0	1
1	1	0

Ad esempio, la riga in giallo di questa tavola della verità del NAND ci dice che:

- *a condizione* che entrambi gli ingressi (IN1, IN2) siano ad 1, cioè al *positivo* di alimentazione
- *allora* la sua uscita (OUT) andrà a *zero*, cioè ad una tensione molto bassa

La tavola della verità di solito elenca tutte le *condizioni* che si possono verificare: entrambi gli ingressi a uno, entrambi gli ingressi a zero, il primo ingresso a zero mentre il secondo è a uno e viceversa. Per ogni

*condizione* la tavola ci indica quale sarà l'effetto sulle uscite dell'integrato, specificando se si disporranno a livello alto (1) oppure a livello basso (0).

## Tavola degli stati = tavola della verità + stati

Le tavole della verità si prestano bene a descrivere il funzionamento della maggior parte dei circuiti logici, con piccole modifiche.

**Gli stati sono una delle aggiunte più utili**, e sono utilizzati anche dai **flip-flop** da tutti gli **integrati logici** di una certa complessità. Capita spesso infatti che la **stessa** condizione porti a risultati **differenti** a seconda dello **stato** in cui si trova un dispositivo. Ritroviamo gli stati in un grandissimo numero di automatismi della vita di tutti i giorni.

Ad esempio:

- lo stessa *condizione* in ingresso = la pressione di un interruttore
- ha due effetti opposti = l'accensione o lo spegnimento di una luce
- a seconda dello *stato* in cui ci si trova = luce spenta o luce già accesa

In un circuito a stati, il valore delle uscite non dipende *soltanto* dalle condizioni che si verificano sugli ingressi (come nel NAND), ma **anche** dallo *stato* in cui ci si trova in quel preciso istante. Quindi nello scrivere la tavola della verità di un interruttore, dovremo necessariamente inserire anche lo stato. La tavola della verità che otteniamo (detta più propriamente *tavola degli stati*) è la seguente:

TABELLA DELL'INTERRUTTORE		
STATO (E USCITA)	INGRESSO	STATO SUCCESSIVO
SPENTO (USCITA= 0)	PREMUTO	ACCESO
	NON PREMUTO	SPENTO
ACCESO (USCITA=1)	PREMUTO	ACCESO
	NON PREMUTO	SPENTO

Notate che in questa tavola abbiamo scritto, fra parentesi, le uscite nella stessa casella dello stato. In gran parte dei casi pratici le uscite dipendono esclusivamente dallo stato in cui ci si trova, per cui è comodo scrivere assieme lo stato e le uscite. Se si deve cambiare una delle uscite, allora si cambia anche stato. Si può anche cambiare stato senza variare in alcun modo le uscite, infatti due stati diversi possono tranquillamente dare le stesse uscite. Un'altra peculiarità di questa tavola è l'introduzione della colonna che riporta lo stato successivo, cioè lo stato in cui ci troveremo non appena la *condizione* di ingresso si verificherà.

**Esempio:** il **contatore** è uno fra i tantissimi integrati che funzionano a stati. La stessa *condizione* di ingresso, rappresentata dall'arrivo di un impulso, otterrà un effetto diverso delle uscite a seconda dello *stato* del contatore. Il valore binario delle uscite dipenderà dal **conteggio raggiunto** dal contatore fino a quel momento. In altre parole, il conteggio è anche lo *stato* del contatore. Per il contatore, così come accadeva per l'interruttore, è lo stato a determinare quali debbano essere le uscite a '0' a quali a '1'.

In un integrato a stati è importante sapere **da quale stato cominciare**. Un buon contatore deve iniziare a contare da zero, non da un numero scelto a caso! Lo stato iniziale, detto anche *stato zero*, è sempre indicato dal costruttore, oppure deve essere presente un piedino apposito che riporta forzatamente l'integrato allo *stato zero*.

## La tavola di un Nutchip

Come accade per moltissimi integrati digitali, anche per i Nutchip lo *stato* determina le *uscite*: i Nutchips sono integrati a stati. I Nutchips hanno quattro uscite: le uscite cambiano solo se *anche* lo stato cambia, per cui accanto ad ogni stato indicheremo anche la rispettiva combinazione di uscite. Per ogni stato, c'è la massima libertà nello stabilire quali uscite debbano essere a 0 e quali a 1.

Ecco qui una tipica tabella di un Nutchip:

state	output 1...4	input 1...4	remote	next	
st00	0 0 0 0	0 - - -	... - - - - key 1	st01 st02	pulsante premuto, azioniamo il timer ordine di accensione dal telecomando
st01	1 0 0 0	timeout 1 min		st00	dopo un minuto spegniamo le luci
st02	1 0 0 0	- - - -	key 2	st00	ordine di spegnere dal telecomando

*Tipica tavola della verità di un Nutchip; st00 è un'abbreviazione per 'stato 00', etc.  
Si tratta di un timer per luci da giardino, luci azionabile da un interruttore o da telecomando.*

Trascuriamo per il momento la colonna *remote* (che vedremo più avanti), e notiamo che abbiamo messo in evidenza le uscite utilizzando un'apposita colonna *output 1...4*. E' in queste caselle che si mettono i valori che devono assumere le quattro uscite del Nutchip, stato per stato. Gli stati possono avere anche più di una condizione di ingresso (è il caso dello stato 0 del nostro esempio, che ne conta due), ma sempre e solo *una* combinazione di uscite. Le uscite non devono necessariamente cambiare fra uno stato e l'altro: nella tabella del nostro esempio l'uscita 1 va a zero nello stato 0, e va a 1 negli stati 2 e 3. Le uscite 2, 3 e 4 invece non cambiano mai.

Rispetto alle tavole viste in precedenza, con i Nutchip possiamo anche inserire un trattino '-' quando non ci interessa specificare il valore di un **ingresso**. Il trattino '-' equivale alla X (*don't care*) che compare talvolta nella descrizione degli integrati commerciali. Un ingresso con il trattino è come che non esistesse e viene ignorato durante la valutazione della condizione.

La colonna *remote* si riferisce al **telecomando**: è un ingresso a tutti gli effetti, che invece di '0' o '1' può assumere i valori *key1*, *key2*, *key3*, *key4*, *key5* e *key6* corrispondenti ai 6 tasti del telecomando. Inserendo i trattini '---' il telecomando viene ignorato. Maggiori dettagli sui telecomandi: [cliccare qui](#).

I Nutchip contengono anche un preciso **temporizzatore**: per utilizzarlo lo inseriamo nella tavola sotto forma di *timeout*. Il *timeout* è una *condizione* che si verifica quando è passato il tempo caricato nel temporizzatore. Lo stato *st01* del nostro esempio è un *timeout* che si verifica dopo un minuto. Maggiori dettagli sui *timeout*: [cliccare qui](#).

## Come si legge la tavola

Gli stati sono separati da una linea orizzontale, quindi questa tavola conta tre stati: *st00*, *st01*, *st02*.

Ogni riga della tabella contiene una *condizione*.

Quindi nel nostro caso lo stato *st00* ha 2 condizioni, mentre *st01* e *st02* contengono una sola condizione.

All'inizio ci si trova nello stato *st00*. Le uscite cambiano in base allo stato. Si analizzano le sole condizioni dello stato in cui ci si trova: se una condizione si verifica\*, ci si porta sullo stato indicato nell'ultima colonna (*next*).

*Nota: (\*)Può capitare che due o più condizioni si avverino nello stesso momento: in tal caso si passa allo stato indicato da quella che viene per prima sulla tavola.*

- La prima colonna riporta il numero dello stato, esempio *st00*, *st01*, *st02*, *st03*, *st04*, etc.  
Si parte sempre da zero.
- Subito dopo lo stato c'è il valore assunto dalle uscite OUT1, OUT2, OUT3 e OUT4 quando ci si trova in quello stato.  
Per ogni stato si inserisce una sola combinazione di valori.
- La terza colonna, assieme alla successiva, forma la *condizione* che si deve verificare per effettuare un cambiamento di stato.  
Qui si specifica il valore necessario sugli ingressi IN1, IN2, IN3 e IN4 perchè la condizione sia soddisfatta.  
Ogni ingresso può essere a '0' (massa), '1' (positivo) o '-' (indifferente, l'ingresso viene ignorato)  
Per ogni stato si possono inserire tutte le condizioni che si vuole; le condizioni si inseriscono cliccando l'apposito bottone.
- In alternativa può comparire la scritta "timeout", nel qual caso la condizione si verifica *allo scadere* del tempo indicato, *qualunque sia* il valore degli ingressi e/o del telecomando.  
Si può inserire *un solo* timeout per ogni stato, cliccando sull'apposito bottone.
- La quarta colonna specifica uno dei 6 pulsanti del telecomando. I pulsanti sono indicati come *key1 ... key6*.  
La condizione si verifica quando il pulsante effettivamente premuto corrisponde a quello indicato nella casella.  
Questa colonna si può usare anche in combinazione con gli ingressi: altrimenti si impostano gli ingressi IN1...IN4 su "indifferente" ('- - -')
- L'ultima colonna indica lo stato successivo, ovvero lo stato in cui portarsi quando la condizione si verifica.  
Se nessuna condizione dovesse essere vera, si rimane nello stato in cui ci si trova.

state	output 1...4	input 1...4	remote	next
st00	0 0 0 0	0 - - -	...	st01
		- - - -	key 1	st02
st01	1 0 0 0	timeout 1 min		st00
st02	1 0 0 0	- - - -	key 2	st00

## Una tavola personalizzata

I Nutchip si **differenziano** dai circuiti integrati logici standard perchè la loro tabella della verità non è fissata in fabbrica: siete voi stessi a "*caricare*" al loro interno la vostra tavola personalizzata. Lo stesso Nutchip può fare mille cose differenti, a seconda della tavola che carichiamo al suo interno. Quando una tavola della

verità non ci piace più, possiamo sempre ricaricare il Nutchip con una tavola nuova, che sostituirà automaticamente quella precedente.

*Per permettervi di provare subito il vostro Nutchip appena acquistato, lo forniamo precaricato con una tavola che lo fa funzionare da lampeggiatore; basta collegare un LED per avere la certezza che il vostro prototipo funziona. In più, se collegate anche un ricevitore ad infrarossi, la stessa tavola vi permette di controllare se il vostro telecomando è compatibile, bloccando il lampeggio ogni volta che riceve un impulso.*

Per **scrivere** la nostra tavola utilizziamo un comune PC ed un software apposito, **Nutstation**. La memorizziamo poi nel chip usando una semplice [interfaccia](#) da collegare alla porta seriale del computer. Una volta programmato, il Nutchip obbedirà fedelmente alla tavola della verità che gli abbiamo inserito sino a che non lo riprogrammeremo con una tavola diversa. Lo stesso chip si può **riprogrammare** migliaia di volte.

*Avere un solo Nutchip è come avere un'intero campionario di integrati, e molto di più. Potremo realizzare anche quelle funzioni logiche che, non essendo standardizzate, richiedono tre o quattro integrati per essere realizzate. In più ogni NUTCHIP contiene, oltre alle porte logiche, anche un [decodificatore per telecomandi](#), un [timer](#) ed un [comparatore analogico](#)!*

## Nutstation, pulsanti e comandi

Prima di tutto, installate il software seguendo le apposite [istruzioni](#).

Tavola verità | Prog. chip

*I due "tab" di Nutstation*

Nutstation è composto da due pagine: una serve per creare e modificare la tavola della verità, l'altra serve per la programmazione del chip. Per passare da una pagina all'altra si clicca sui "tab".



Nella pagina "[Tavola verità](#)" si creano e modificano le tavole della verità...









....poi nella pagina "[Programmazione chip](#)" si trasferisce la tavola sul chip (attraverso la porta seriale del PC)




..se usate il telecomando, prima di programmare il chip dovreste scegliere i codici da utilizzare per ogni tasto: cliccate sul pulsante "[Telecomando](#)".

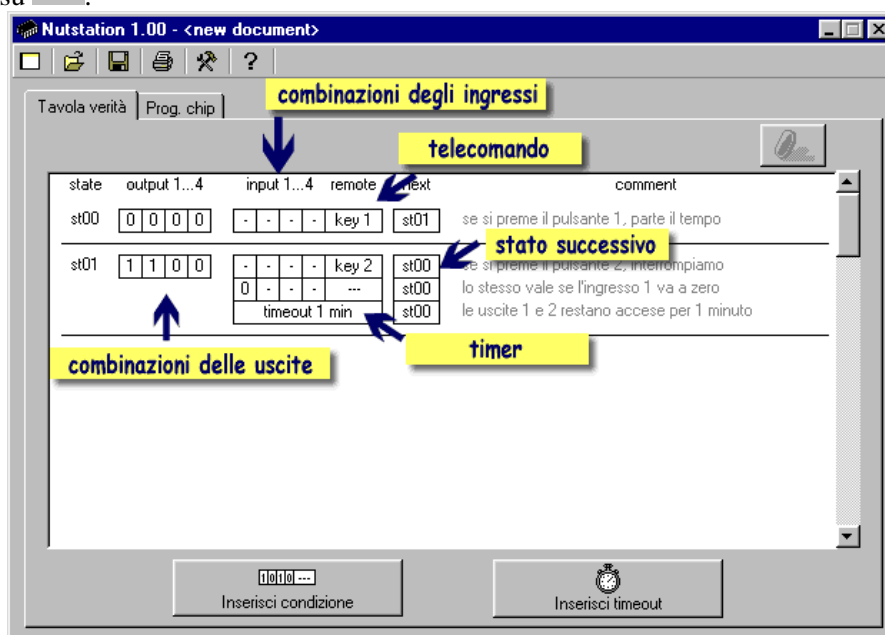


Per salvare o stampare il vostro lavoro usate i pulsanti della **toolbar** (in alto a sinistra):

-  Nuova tavola: cancella la tavola corrente e predisporre i valori standard per un nuovo lavoro.
-  Apri file: abbandona la tavola corrente e carica al suo posto una tavola memorizzata su disco (files .nut).
-  Salva con nome: crea una copia della tavola corrente sul disco.
-  Stampa la tavola corrente sulla stampante predefinita da Windows.
-  Setup: imposta la porta seriale da utilizzare pre programmare il chip
-  Guida: mostra la guida di Nutstation e dei Nutchip

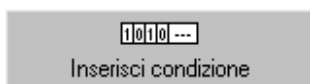
## Tavola della verità

All'avvio Nutstation carica automaticamente l'ultima tavola salvata. Se si vuole lavorare con una tavola nuova, fare clic su .



La riga corrente è evidenziata in giallo. Se qualche valore **diventa rosa**, allora avete commesso un errore: ad esempio avete indicato uno stato che non esiste..

Si **cambia** il valore delle caselle cliccandoci sopra.: ad esempio, cliccando sulla casella di un'uscita il valore cambierà da 0 a 1 e viceversa. Per lavorare su un'altra riga basta portarvi sopra e cliccare con il mouse. Per inserire un commento invece basta digitarlo sulla tastiera.



Con questo pulsante si aggiungono nuove righe (condizioni) alla tavola. Le righe di condizione contengono una *condizione logica* degli ingressi. La nuova condizione viene aggiunta allo stato evidenziato in giallo al momento del click. Se si clicca quando nessuno stato è evidenziato, cioè quando è evidenziata una riga vuota, allora si crea automaticamente anche un nuovo stato.



Con questo pulsante si aggiungono nuove righe (timeout) alla tavola. Le righe di *timeout* contengono un tempo scaduto il quale si passa allo stato successivo indicato. Si può specificare un solo timeout per ogni stato, e il pulsante è disattivo se lo stato corrente ha già un timeout. Se la riga corrente non appartiene a nessuno stato, verrà creato anche un nuovo stato, altrimenti il timeout verrà accordato alle condizioni dello stato corrente.



Il pulsante gomma cancella in modo definitivo la riga corrente.

Per ulteriori dettagli su come è fatta la tavola della verità [cliccare qui](#).

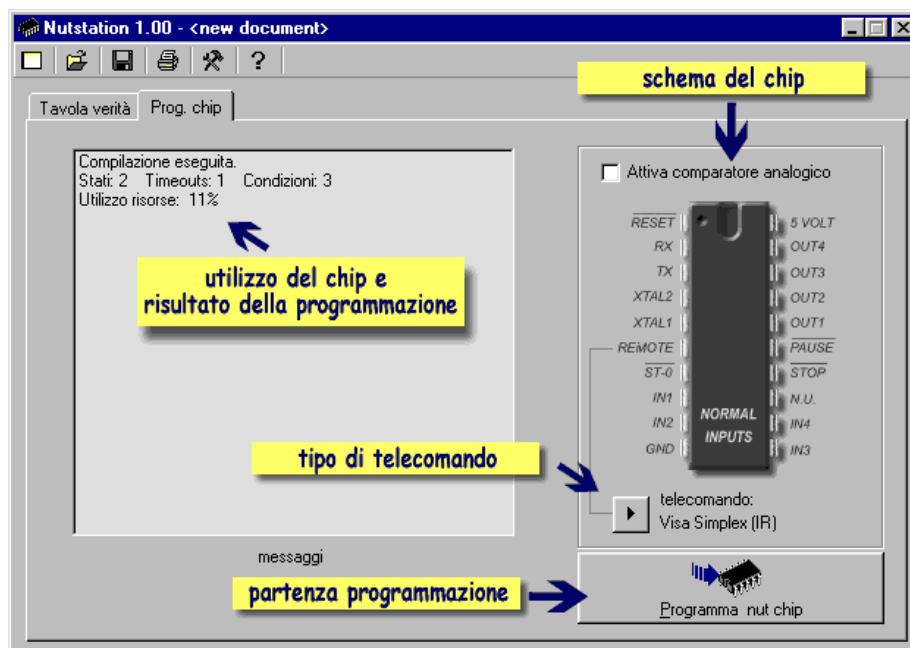
## Programmazione di un Nutchip

Tavola verità | Prog. chip

Cliccando sul tab "Prog. chip" si passa alla pagina di programmazione del chip. A sinistra compare il riquadro con lo stato di programmazione: qui sono elencati compaiono gli eventuali messaggi di errore.

Per programmare un Nutchip vi servono:

- una porta seriale disponibile sul PC
- una tavola della verità funzionante. Nutstation si rifiuta di programmare tavole errate.
- [l'interfaccia](#) che converte i segnali del PC a livello logico di 0...5V utilizzato dal Nutchip
- un circuito per fornire l'alimentazione, il clock, il reset al Nutchip. Può essere il circuito definitivo del vostro progetto o qualsiasi altro [semplice circuito](#) che usa il Nutchip.




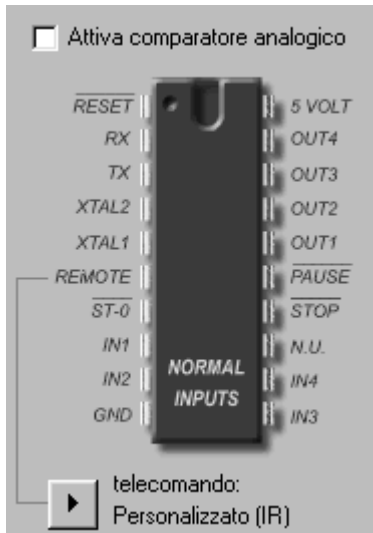
Lo stesso chip si può riprogrammare più di mille volte. La programmazione non richiede nè cancellazione e nè formattazione.

Una volta programmati, i Nutchip mantengono la loro programmazione anche se vengono spenti, tolti dal circuito, saldati, spediti, etc.



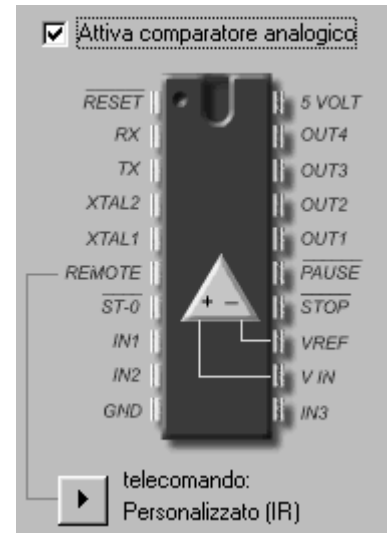
Per programmare il Nutchip cliccare questo pulsante. **Se la tavola della verità contiene degli errori o è incompleta, questo pulsante si disattiva automaticamente** e la programmazione non si può fare!

E' ovviamente necessario che il circuito che ospita il Nutchip sia alimentato con 5 volt e collegato alla porta seriale del PC attraverso l'interfaccia. La porta seriale del PC si sceglie nel menù di setup (tasto )



Alcuni progetti sfruttano il [comparatore analogico](#) del Nutchip, altri invece non ne fanno uso per cui IN4 funziona come un normale ingresso digitale.

Cliccate su **Abilita comparatore analogico** per attivare o meno il comparatore interno del Nutchip. Il comparatore confronta la tensione presente sul pin Vin (IN4) con la tensione sul pin Vref, ed il risultato sostituito al valore logico di IN4. Se caricate un file già pronto, le opzioni saranno regolate **automaticamente**; partendo invece da una nuova tavola, l'ingresso è preregolato come ingresso digitale (IN4). In caso di dubbi lasciatelo così.



## Telecomando



Sulla pagina "Programmazione chip" c'è il promemoria del tipo di telecomando usato. I codici restano memorizzati automaticamente, perciò se usate sempre il medesimo telecomando non avrete bisogno di selezionarlo ogni volta. Per selezionare un diverso tipo di telecomando fate click sul pulsante a freccina e si aprirà la pagina del telecomando. Se il vostro progetto non utilizza il telecomando, potete ignorare questa selezione.

Per ogni telecomando compare una fotografia indica le corrispondenze dei tasti da key1 a key6.

Potete scegliere uno dei telecomandi standard, cliccando su:

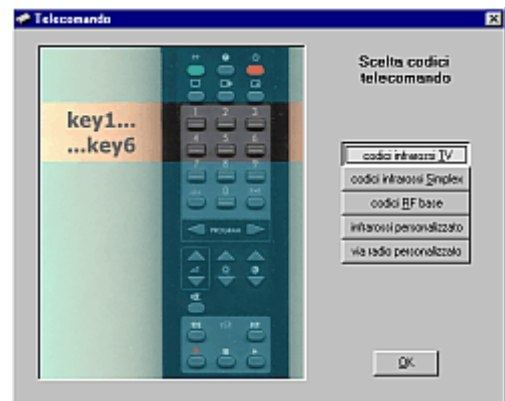
- codici infrarossi TV
- codici infrarossi Simplex (preprogrammato come "Gruppo Philips")
- codici RF base

oppure personalizzare i codici di un telecomando infrarossi o RF, cliccando su:

- infrarossi personalizzato
- via radio personalizzato

Cliccare qui per [maggiori informazioni](#) sui codici e sui telecomandi supportati.

Nota: il telecomando virtuale Nutchip Commander funziona sempre, indipendentemente dal tipo di telecomando selezionato.



## Telecomandi personalizzati



Scegliendo uno dei due telecomandi personalizzati (infrarossi o RF) si ha la possibilità di assegnare ad ognuno dei pulsanti un codice a piacere. Basta portare il mouse sulla figura del telecomando e fare click sul tasto di cui si vuole cambiare il codice per fare comparire la finestra "codice tasto":




Se non si conosce il codice di un tasto, lo si può ricevere dal Nutchip stesso tramite la funzione di autoapprendimento.

- Se il circuito con il Nutchip è collegato al PC, il tasto "Apprendi" si attiva automaticamente.
- Premete il tasto sul telecomando in modo che il Nutchip lo riceva.
- Poi fate click su "Apprendi" ed automaticamente ed il codice appena ricevuto dal Nutchip comparirà nella finestrella "codice" al posto del numero precedente.

## Selezione della porta seriale ed interfaccia PC



Per programmare il chip è necessario collegare l'interfaccia seriale al PC su una delle porte seriali disponibili. Facendo clic su  compare la finestra di setup da cui selezionerete la porta seriale da voi utilizzata. Se una porta non è selezionabile (come com3 e com4 in questo esempio) significa che è già utilizzata da altre risorse oppure non è disponibile.

Per dettagli sulla realizzazione dell'interfaccia e del cavo di collegamento [cliccare qui](#)

-----