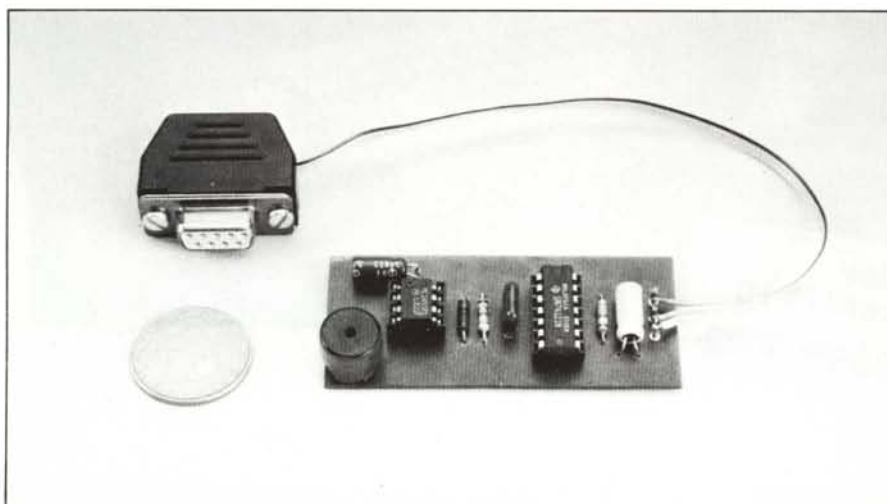




# VIC da zero

di Tommaso Pantuso



## Aggiungiamo un Beep

*La volta scorsa abbiamo cominciato ad esaminare come sfruttare una delle peculiarità più rilevanti del VIC 20, e cioè la possibilità di manipolare le sequenze d'interrupt della macchina introducendo delle routine intermedie utili a modificare il comportamento normale del sistema per asservirlo a scopi specifici. Abbiamo allora costruito un programma molto semplice che produceva come output un beep ogni volta che veniva premuto un tasto dando la conferma acustica di tale pressione. Oggi tratteremo qualcosa di simile ma struttureremo il processo in modo diverso, cioè lo renderemo più funzionale velocizzandone lo svolgimento grazie all'ausilio di alcuni componenti esterni di cui analizzeremo abbastanza in dettaglio le caratteristiche.*

### Il vecchio ed il nuovo

Riassumiamo brevemente i fatti precedenti. Ogni sessantesimo di secondo il VIC interrompe tutte le operazioni in corso e provvede a mandare in esecuzione alcune routine di controllo e di aggiornamento quali quelle di verifica dello stato della tastiera, aggiornamento di un contatore interno, TI, ed altro.

Prima di far questo il sistema provvede a salvaguardare in un'area appositamente concepita i parametri essenziali che gli permetteranno di riprendere l'esecuzione da dove era stata interrotta prima della diramazione verso le routine d'interrupt.

Il meccanismo di salto è il seguente: viene letto il contenuto delle locazioni esade-

cimali 0314 e 0315 effettuando il salto all'indirizzo da esse puntato. Normalmente esse contengono rispettivamente i numeri 191 e 234 (decimale) che nel consueto svolgimento indirizzano alla locazione

$$191 + 234 \times 256 = 60095$$

da cui si parte per la manipolazione dell'interrupt, cioè per l'esecuzione delle consuete routine.

Noi possiamo modificare il contenuto di tali locazioni (0314-0315) per mandare il sistema ad eseguire dei segmenti di programma da noi introdotti alla fine delle quali verrà posto un salto all'indirizzo 60095 per permettere al sistema di riprendere il normale svolgimento delle sue funzioni. Il tutto avverrà sempre alla frequenza di sessanta volte al secondo.

Sfruttando questi fatti abbiamo quindi avuto la possibilità di mandare la macchina ad eseguire un programma di nostra concezione il quale controllava se era stato premuto un tasto e, se la risposta era affermativa, attivava il generatore audio interno al computer, produceva un ritardo alla fine del quale spegneva tale generatore e partiva per le consuete manipolazioni d'interruzione. Il ritardo lo abbiamo ottenuto grazie ad uno dei due timer interni al VIA 6522 contenuto nel VIC 20 il quale, caricato con un numero N da noi scelto, viene decrementato, dopo l'avvio del conteggio, di un'unità circa ogni 1.1 microsecondi. Il ritardo da noi introdotto è quindi di tipo hardware perché sfrutta essenzialmente le possibilità offerte dal chip interno alla macchina precedentemente menzionato.

Abbiamo osservato però che durante lo scaricamento del contatore, il microprocessore non fa nient'altro per parecchi millisecondi e ciò rallenta la velocità d'uso della tastiera a volte in maniera intollerabile. Infatti per ottenere un beep "pulito" bisogna tenere l'oscillatore in funzione per un tempo superiore almeno a 100 millisecondi. Per ovviare a questo inconveniente possiamo strutturare il sistema in modo diverso costruendo un circuito esterno controllato dal computer tramite un impulso di attivazione di brevissima durata (anche pochi microsecondi) evitando così di portar via tempo utile al microprocessore che verrebbe impegnato solo per tale periodo. Tale circuito deve contenere un oscillatore che generi la nota per il beep ed un generatore di ritardo per controllare la durata del tempo di oscillazione della nota stessa.

Un sistema del genere è schematizzato in figura 1 ed opera nella maniera seguente: ogni volta che viene premuto un tasto; un segnale rappresentato da una variazione di livello su una delle uscite del computer a noi accessibili, è inviato al generatore di ritardo che viene attivato; quest'ultimo a sua volta controlla tramite una sua linea l'oscillatore tenendolo attivo durante il tempo in cui è attivo il ritardo. Il funzionamento dettagliato sarà spiegato tra breve. Prima però vogliamo analizzare il modo di

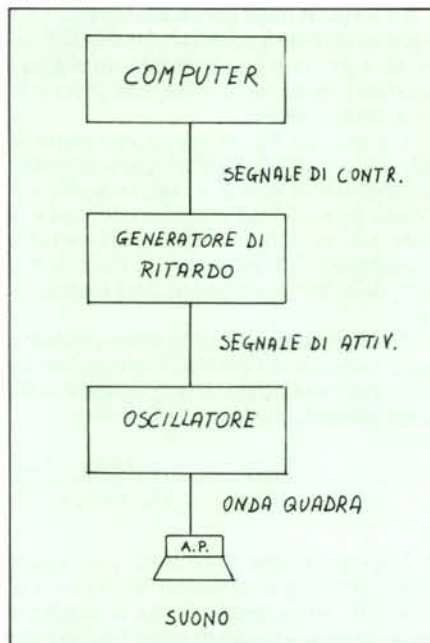


Figura 1 - Il segnale in uscita dal computer sotto forma di un impulso viene applicato in ingresso al monostabile generatore di ritardo. L'uscita di quest'ultimo allora cambierà stato per tutta la durata del ritardo permettendo in questo intervallo all'oscillatore audio di produrre la nota sotto forma di un'onda quadra che sarà "rivelata" dall'altoparlante A.P.

operare e le caratteristiche dei componenti elettronici che ci permetteranno di realizzare fisicamente la nostra idea.

### Il generatore di ritardo

Sono molti i modi in cui si può realizzare un ritardo hardware tramite dei componenti elettronici e tra tutti ne abbiamo scelto uno che pensiamo sia di immediata comprensione e di facile applicazione perché ottenuto tramite un componente programmabile: il *multivibratore monostabile* 74121, riportato in figura 2. Il monostabile è un elemento che possiede un *solo stato stabile*, nel senso che la sua uscita si trova normalmente o a livello basso (per esempio 0 volt) o a livello alto (per esempio 5 volt) e che può portarsi, se sollecitato, nello stato opposto solo per un periodo di tempo limitato ritornando, al termine di tale periodo, nella sua naturale condizione di equilibrio. Questo processo è schematizzato in figura

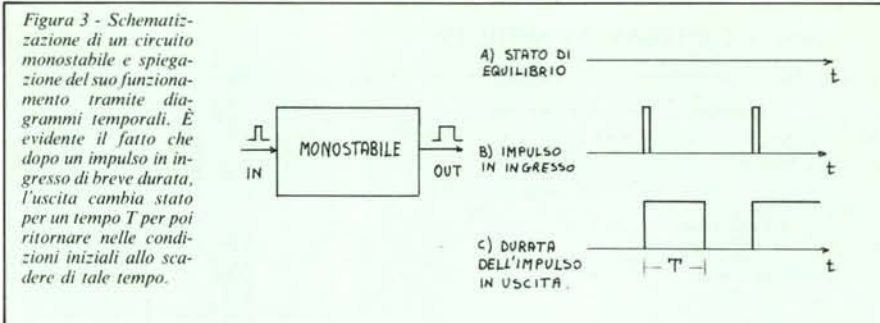
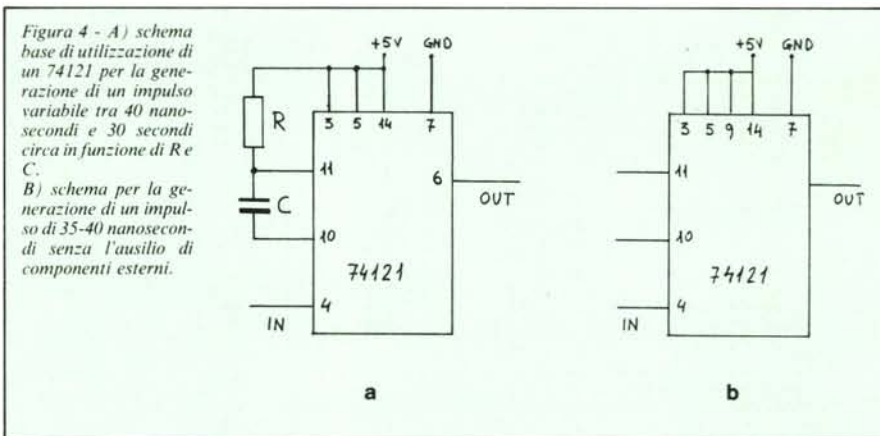


Figura 3 - Schematizzazione di un circuito monostabile e spiegazione del suo funzionamento tramite diagrammi temporali. È evidente il fatto che dopo un impulso in ingresso di breve durata, l'uscita cambia stato per un tempo T per poi ritornare nelle condizioni iniziali allo scadere di tale tempo.



3 dove osserveremo che l'uscita del monostabile è nel suo stato normale a livello basso. Quando sopraggiunge in ingresso un impulso di comando (diagramma B), lo stato dell'uscita si porta a livello alto e vi permane per un tempo T per poi ritornare nelle condizioni d'origine.

Si noti che la larghezza (cioè la durata) dell'impulso mandato in ingresso è molto più breve di quella dell'impulso d'uscita.

In particolare non è sempre necessario conoscere la durata del segnale di comando essendo la durata di quello d'uscita indipendente da essa; si può allora affermare che tramite un monostabile si possono ottenere impulsi di durata nota, quelli da noi programmati, partendo da impulsi di durata sconosciuta.

La durata del ritardo è regolata da due componenti passivi esterni e cioè un condensatore C ed un resistore R che per il 74121 vengono collegati sui piedini 10, 11 e 14 come indicato in figura 4a.

C può variare in un campo compreso tra 10 picofarad e 10 microfarad, mentre R spazia tra gli estremi di 2 kohm e 40 kohm.

Con l'appropriata scelta di tali componenti l'impulso in uscita può variare tra 40 nanosecondi e 28 secondi, secondo la relazione

$$T = CR \ln 2 \approx 0.7 CR$$

dove ln sta per logaritmo naturale.

Piccoli ritardi, dell'ordine 30-35 nanosecondi, possono essere ottenuti senza l'ausilio di componenti esterni collegando il chip come rappresentato in figura 4b.

Aggiungiamo che il monostabile in questione possiede due uscite che assumono stati logici opposti, quindi indicando lo stato di una viene automaticamente individuato quello dell'altra.

### L'oscillatore audio

Anche un generatore di nota può essere realizzato in moltissimi modi e tra questi noi abbiamo scelto l'uso di un componente che ha avuto molta fortuna nel campo della progettazione per la sua eccezionale versatilità che lo mette in grado di essere utilizzato in un numero molto elevato di applicazioni. Stiamo parlando del chip *timer* 555 introdotto per la prima volta sul mercato dalla Signetics Corporation SE 555/NE 555 ed in seguito prodotto secondo le loro versioni da uno svariato numero di case costruttrici (RCA, National, Motorola, TI, Fairchild ecc.) negli ultimi 13 anni (figura 5 a pagina 124).

La versatilità di tale elemento è paragonabile a quella di un amplificatore operazionale. Non possiamo illustrare le centinaia di applicazioni ottenibili da esso, né

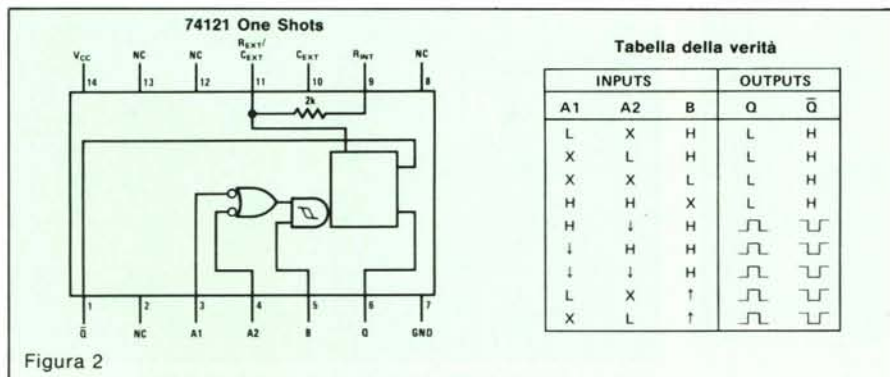


Figura 2

## CIRCUITO INTEGRATO LINEARE 555

Caratteristiche elettriche (TA = 25°C, VCC = +5V to +15 unless otherwise specified)

PARAMETER	TEST CONDITIONS	SE 555			NE 555			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Supply Voltage		4.5		18	4.5		15	V
Supply Current	VCC = 5V RL = ∞		3	5		3	6	mA
	VCC = 15V RL = ∞		10	12		10	15	mA
Timing Error	Low State, Note 1							
	RA, RB = 1KΩ to 100KΩ							
	C = 0.1 μF Note 2							
Initial Accuracy			0.5	2		1		%
Drift with Temperature			30	100		50		ppm/°C
Drift with Supply Voltage			0.005	0.02		0.01		%/Volt
Threshold Voltage			2/3			2/3		V
Trigger Voltage	VCC = 15V		4.8	5		5		V
	VCC = 5V		1.45	1.67		1.67		V
Trigger Current			0.5			0.5		μA
Reset Voltage			0.4	0.7		0.7	1.0	V
Reset Current			0.1			0.1		mA
Threshold Current			0.1	0.25		0.1	0.25	μA
Control Voltage Level	VCC = 15V		9.6	10		9.0	10	V
	VCC = 5V		2.9	3.33		2.6	3.33	V
Output Voltage Drop (low)	VCC = 15V							
	ISINK = 10mA		0.1	0.15		0.1	0.25	V
	ISINK = 50mA		0.4	0.5		0.4	0.75	V
	ISINK = 100mA		2.0	2.2		2.0	2.5	V
	ISINK = 200mA		2.5			2.5		V
Output Voltage Drop (high)	VCC = 5V							
	ISOURCE = 8mA		0.1	0.25		0.1	0.25	V
	ISINK = 5mA					0.25	0.35	V
Rise Time of Output	ISOURCE = 200mA		12.5			12.5		V
	VCC = 15V							
Fall Time of Output	ISOURCE = 100mA		13.0			13.3		V
	VCC = 5V		3.0	3.3		2.75	3.3	V
			100			100		nssec
			100			100		nssec

## NOTES:

- Supply Current when output high typically 1mA less.
- Tested at VCC = 5V and VCC = 15V.
- This will determine the maximum value of RA + RB for 15V operation, the max total R = 20 megohm.

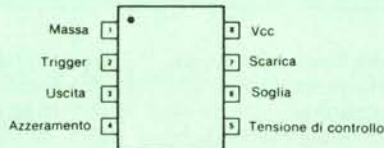


Figura 5

spiegarne molto dettagliatamente il funzionamento essendo ciò estraneo agli scopi del presente articolo; ci limiteremo perciò ad illustrare il funzionamento del circuito da noi utilizzato ed a dare i fondamentali elementi per il calcolo delle caratteristiche. Come curiosità aggiungiamo che un 555 è utilizzato nel VIC 20 per realizzare il circuito di reset per i chip della macchina all'accensione.

Il timer in questione può essere utilizza-

to sia come elemento monostabile, quindi come generatore di ritardi, sia come elemento bistabile, cioè come sistema la cui uscita possiede due stati stabili e che può passare da uno all'altro se sollecitato da un impulso di comando, permanendovi fino ad una nuova sollecitazione, sia come elemento astabile, cioè senza uno stato logico stabile poiché l'uscita varia continuamente tra il livello alto e quello basso. Noi usere-  
mo il 555 in quest'ultima configurazione.

La forma d'onda rilevata in uscita da un circuito astabile ha l'andamento della figura 6b e per tale caratteristica un sistema astabile prende anche il nome di generatore di onda quadra.

L'uscita del 555 possiede una potenza tale da poter pilotare un altoparlante quindi, inviando ai capi di un tale componente l'onda generata dall'astabile, udiremo una nota ad un volume abbastanza elevato. L'ampiezza dell'onda varierà tra 0 volt e Vcc, dove Vcc è la tensione di alimentazione.

Il collegamento del chip come astabile è rappresentato in figura 6a e riferendoci ad essa, potremo ricavare la frequenza della nota generata mediante la formula

$$f = \frac{1}{\ln 2 (R_1 + 2R_2)C} \approx \frac{1.443}{(R_1 + 2R_2)C} \text{ Hz.}$$

L'onda quadra generabile può essere asimmetrica cioè il tempo in cui essa si mantiene alta e quello in cui si mantiene bassa possono essere di durata differente. Definiamo *duty cycle* di un'onda del genere il rapporto tra il tempo in cui il segnale permane nello stato alto e la durata dell'intero ciclo. Un esempio di segnali con differenti *duty cycle* è dato in figura 7.

Se  $t$  è il tempo in cui il segnale resta a livello alto e  $T$  è il periodo completo, DC può essere ricavato dalla relazione

$$DC = \frac{t}{T} = \frac{R_1 + R_2}{R_1 + 2R_2} \text{ secondi}$$

Nella figura 6a si può osservare che il pin 4 è indicato come terminale di reset per il chip. La sua funzione nella configurazione circuitale da noi sfruttata è quella di interdire l'oscillatore ogni qualvolta esso è portato a livello basso. In altre parole, se tale terminale viene lasciato libero o collegato a Vcc, dall'altoparlante percepiremo la nota generata dall'astabile, mentre se viene collegato a massa, cioè a livello basso, non sarà più generata alcuna nota.

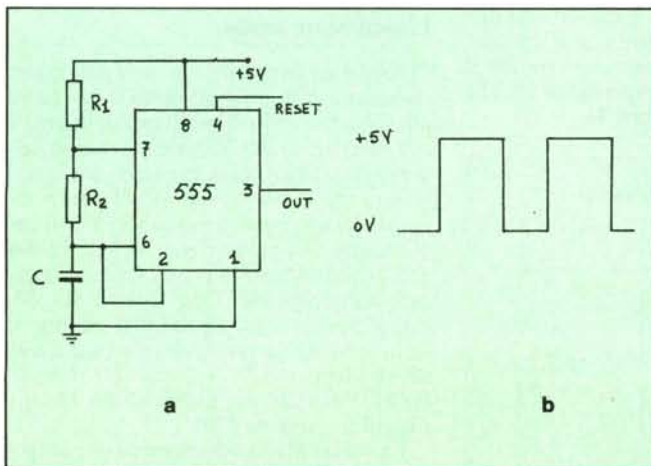


Figura 6 - Schema di un 555 collegato come generatore di onda quadra e possibile forma d'onda rilevabile sulla sua uscita. Se il pin 4 è posto a massa l'oscillatore si blocca per tutto il tempo in cui tale pin permane nel livello basso.

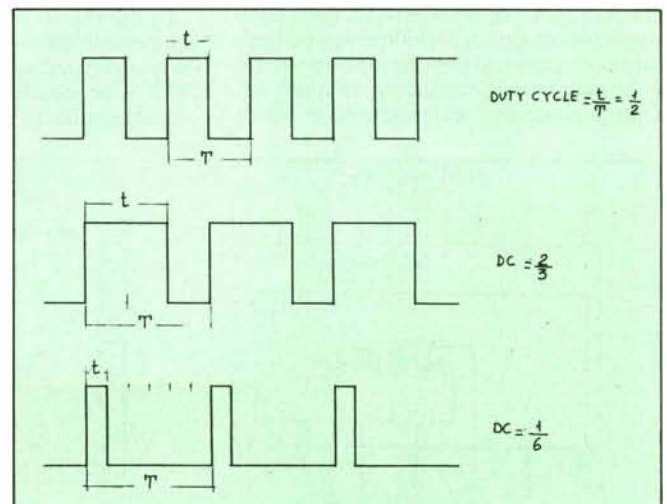
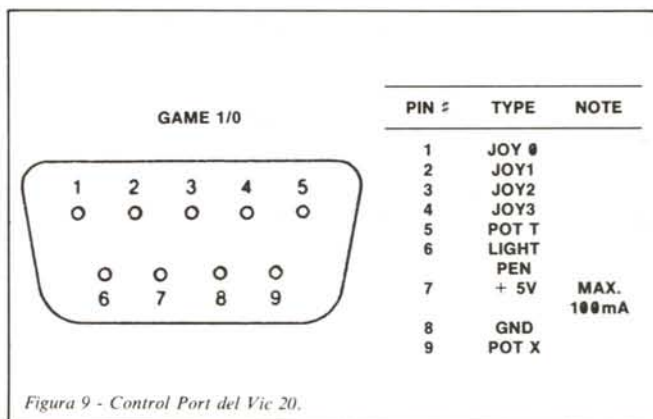
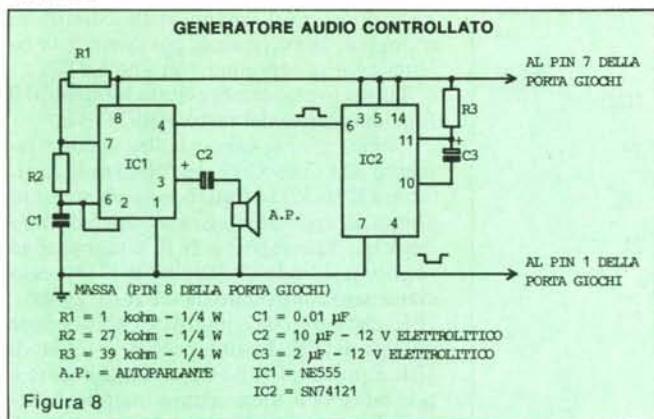


Figura 7 - Vari tipi di onda quadra generabile tramite un astabile.

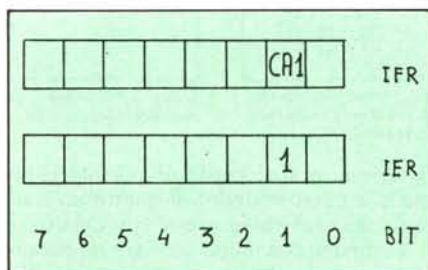
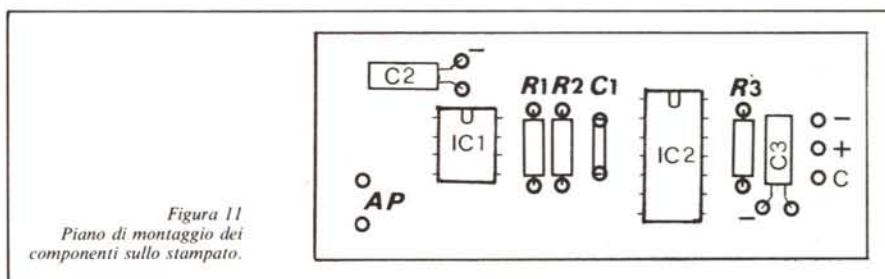
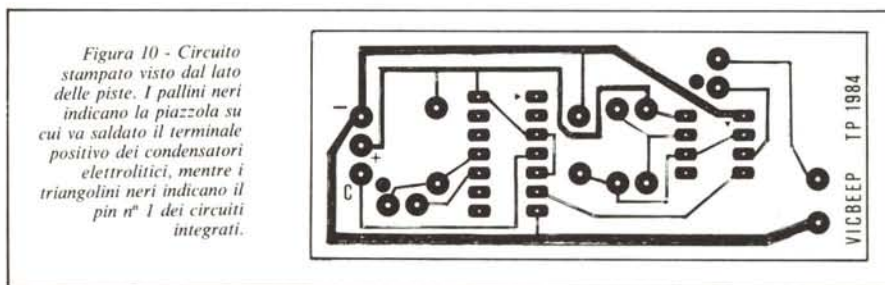


## Il circuito completo

Nella figura 8 presentiamo lo schema completo del generatore di nota controllato di cui diamo una breve descrizione del funzionamento. Il segnale di comando proveniente dal computer viene prelevato da una linea di uno dei VIA interni al VIC 20 e precisamente da PA7. Tale segnale è generato mandando questa linea a livello basso per un istante e riportandola subito dopo nelle condizioni iniziali: avremo così ottenuto un impulso negativo.

Il piedino 4 del monostabile 74121 è collegato al terminale numero 1 della porta giochi e mantenuto normalmente a livello alto (figura 9). In questa condizione l'uscita di tale elemento, cioè il pin 6, si trova a livello basso ed essendo collegata al terminale di reset del 555 mantiene interdetto l'oscillatore. Quando il terminale 4 di IC2 riceve l'impulso negativo proveniente dal computer, l'uscita 6 si porta a livello alto per circa 80 millisecondi (calcolati in base ai valori di R3 e C3 come spiegato), abilitando il funzionamento del generatore di nota per la durata di questo intervallo, il quale genererà un segnale di frequenza 2600 Hz circa (calcolato sempre con una delle formule fornite). Se si vuole contenere il circuito in uno spazio estremamente ridotto conviene utilizzare un microaltoparlante del tipo Souducere con diametro di circa 1 centimetro.

Per chi volesse realizzare il circuito descritto forniamo in figura 10 il disegno dello stampato ed in figura 11 il piano di assemblaggio dei componenti sullo stampato. La reperibilità del materiale elettronico non dovrebbe creare difficoltà, trattandosi di elementi molto comuni e notevolmente diffusi. Per il collegamento del circuito alla porta giochi si deve usare un connettore Cannon a 9 poli connettendo il terminale C indicato sul C.S. al polo numero 1 e quelli contrassegnati da un "+" ed un "-" rispettivamente ai poli numero 7 ed 8. Non ci sono accortezze particolari da menzionare per la realizzazione pratica del circuito, se non l'avvertenza di non invertire la polarità dei condensatori elettrolitici od il senso degli integrati durante l'inserimento sullo stampato. Inoltre sfruttando le formule che vi abbiamo fornito potrete modificare i



valori delle resistenze o dei condensatori per variare la frequenza del beep o la sua durata.

## Il software

Il listato 1 (pag. 126) riproduce il programma in LM che mette in funzione il beep. Esso differisce da quello fornito il mese scorso solo nell'ultima sezione rimanendo invariato per il resto. È sistemato a partire dalla locazione decimale 674 ed occupa 38 byte. Ricordiamo che tra le locazioni 673 e 779 il sistema possiede byte liberi contenenti 0 all'accensione.

La prima parte della routine proposta serve per modificare il valore del puntatore alle routine d'interrupt del VIC 20, sostituendone uno che indirizza al nostro programma di beep e su questo abbiamo ampiamente discusso la volta scorsa. Degli spazi liberi in cui è memorizzata l'istruzione NOP parleremo tra breve.

Con la seconda sezione si controlla il contenuto del registro 197 e se questo è diverso da 64 (decimale) si mette in funzione il beep, se no si salta alle normali routine di manipolazione.

Nell'ultima sezione si pone nel Registro Direzione Dati (DDR) di uno dei VIA interni allocato a 37139 la parola binaria 10000000 o 128 decimale o 84 esadecimale, la quale ottiene come effetto quello di mandare a zero la linea PA7 che normalmente è alta. Subito dopo viene memorizzato in DDR il valore iniziale (132 decimale) e la linea ritorna a livello alto.

Il risultato di quest'insieme di operazioni è quello di produrre in uscita l'impulso che andrà a pilotare il monostabile e che ha la durata di circa 13 microsecondi. Come al solito la routine si lancia con una SYS, in questo caso SYS 674, dopo essere stata caricata in macchina. Per il caricamento sfrutteremo il listato 2 che memorizza il

ABILITAZIONE						
SEI	:	DISABILITA GLI INTERRUPT	:	78	: 120	02A2
LDA ## B4	:	BYTE BASSO LOCAZIONE DI	:	A9 B4	: 169 180	
	:	INIZIO DELLA ROUTINE BEEP	:		:	
STA #0314	:	LO PONE NELLA LOCAZIONE	:	8D 14 03	: 141 20 3	
	:	DECIMALE 788	:		:	
LDA ## 02	:	BYTE ALTO LOCAZIONE DI	:	A9 02	: 169 2	
	:	INIZIO DELLA ROUTINE BEEP	:		:	
STA #0315	:	LO PONE NELLA LOCAZIONE	:	8D 15 03	: 141 21 3	
	:	DECIMALE 789	:		:	
NOP	:	SPAZIO LIBERO	:	EA	: 234	
NOP	:	SPAZIO LIBERO	:	EA	: 234	
NOP	:	SPAZIO LIBERO	:	EA	: 234	
NOP	:	SPAZIO LIBERO	:	EA	: 234	
NOP	:	SPAZIO LIBERO	:	EA	: 234	
CLI	:	ABILITA GLI INTERRUPT	:	58	: 88	
RTS	:	RETURN	:	60	: 96	

DISCRIMINAZIONE						
LDA ## 40	:	64 DECIMALE IN A	:	A9 40	: 169 64	02B4
BIT #00C5	:	ESEGUE L'AND DI A CON IL	:	2C C5 00	: 44 197 0	
	:	CONTENUTO DEL REG. 197	:		:	
BNE 02C5	:	SE NON E' SODDISFATTO	:	D0 0A	: 208 10	
	:	SALTA ALL'ULTIMA ISTRUZ.	:		:	

BEEP HARD						
LDA ## 84	:	128 DECIMALE IN A	:	A9 84	: 169 128	
STA #9113	:	LO PONE NEL REGISTRO DEC.	:	8D 13 91	: 141 19 145	
	:	37139 (DDRA) E ABBASSA PA2	:		:	
LDA ## 80	:	132 DECIMALE IN A	:	A9 80	: 169 132	
STA #9113	:	LO PONE NEL REGISTRO DEC.	:	8D 13 91	: 141 19 145	
	:	37139 (DDRA) E ALZA PA2	:		:	
JMP #EABF	:	SALTA ALL'INTERRUPT ORIG.	:	4C BF EA	: 76 191 234	02C5

Listato 1 - Programma per la generazione di un impulso in uscita da una linea di un VIA.

```

10 REM * BEEP HARD *
20 REM
30 FOR I=674T0711
40 READ A:POKE I,A:NEXT
50 DATA 120,169,180,141,20,3,169,2,141,21,3,234,234
55 DATA 234,234,234,88,96,169,64,44,197,0,208,10,169
60 DATA 128,141,19,145,169,132,141,19,145,76,191,234

```

Listato 2 - Programma Basic per l'introduzione in macchina della routine riportata nel listato 1. Dopo il RUN esso può essere cancellato. Prima del RUN, dopo un eventuale caricamento da supporto magnetico, premere insieme RUN/STOP e RESTORE.

programma utilizzando delle POKE; inoltre, come oramai sapete, i blocchi di DATA non sono altro che i codici operativi ed i dati del programma in linguaggio macchina tradotti in decimale per poter effettuare l'introduzione diretta tramite l'istruzione POKE e che troviamo indicati nella colonna più a destra del listato 1.

Avviato il programma la macchina controllerà ogni sessantesimo di secondo se è stato premuto un tasto ed invierà l'impulso programmato in caso affermativo.

Tenete presente che l'idea di aggiungere dei componenti esterni per velocizzare le operazioni è nata perché nel caso di un beep prodotto tramite l'oscillatore interno ed un ritardo ottenuto sfruttando la capacità del timer del VIA si è notata una certa lentezza durante lo spostamento del cursore sullo schermo per effettuare delle correzioni o durante la gestione dei testi. Negli altri casi non c'è nessun problema usando il generatore di beep ottenuto sfruttando solo le capacità del sistema, perché in elab-

borazione non si perde di velocità (solo qualche microsecondo). È questo un concetto che vorremmo avere ben chiaro.

Torniamo ora un po' indietro ad esaminare il perché degli spazi occupati dall'istruzione NOP.

Quando nel programma si incontra questa istruzione, non viene effettuata nessuna operazione (NOP = No Operation) e tutto va come se essa fosse ignorata. Si possono quindi introdurre delle NOP per lasciare degli spazi liberi nei quali possono in seguito essere inserite nuove istruzioni. Di fatto per ogni NOP incontrato il sistema non opera per due cicli macchina, dopodiché esso riprende le sue consuete funzioni. Gli spazi da noi lasciati servono appunto per inserire un nuovo segmento di cui andiamo a spiegare la funzione.

Una volta abilitato, il programma viene disabilitato premendo insieme i tasti RUN STOP e RESTORE. Questa operazione la effettuiamo sempre quasi meccanicamente quando vogliamo arrestare un'elaborazio-

ne e riportare il sistema nelle condizioni d'origine. Siamo quindi poi costretti a rilanciare il programma con una SYS.

Tutto ciò può essere evitato bloccando il funzionamento del tasto RESTORE.

Come saprete, esso è collegato direttamente alla linea CA1 del VIA tra le locazioni 37136-37150 e quando viene premuto abilita un apposito flag nel registro di stato del VIA. Tale registro, IFR, è mappato in memoria all'indirizzo decimale 37149; esso viene però controllato da un altro registro, IER, che permette o interdice l'abilitazione dei flag di IFR. In altre parole, se un bit di IER è posto ad 1, il flag corrispondente a tale bit in IFR viene alzato quando è rilevato l'evento attivante; se lo stesso bit di IER è posto a zero, lo stesso evento non abiliterà il corrispondente flag in IFR. In particolare se noi premiamo il tasto RESTORE provocheremo un cambiamento di stato sulla linea CA1 sopra menzionata e questo evento farà alzare il flag F2 (bit 2) di IFR abilitato appunto a rilevare i cambiamenti di stato su questa linea. L'abilitazione è di fatto permessa dal bit 2 di IER che si trova in condizione 1. Se però portiamo a 0 tale bit, il sistema non si accorgerà più delle transizioni che avvengono su CA1, quindi neanche della pressione del RESTORE.

Su come agire su IER per abilitare o disabilitare gli interrupt abbiamo già discusso abbondantemente in un articolo precedente. Riepiloghiamo però brevemente le operazioni da compiere per mani-

OFF RESTORE					
LDA ## 02	:	2 DECIMALE IN A	:	A9 84	: 169 128
STA #911E	:	LO PONE NEL REGISTRO DEC.	:	8D 1E 91	: 141 30 145
	:	37150 (IER) E DISABILITA	:		:
	:	IL TASTO RESTORE	:		:

Listato 3 - Se si vuole, questa routine può essere introdotta negli spazi dei programmi precedenti occupati da NOP.

polare il registro in questione. Esso è composto da otto bit ed i bit da 0 a 6 permettono o no l'attivazione dei flag di IFR a seconda che in essi sia memorizzato un 1 od uno 0 (si faccia riferimento alla figura 12 a pagina 125).

Il bit 7 ha invece un'altra funzione: se esso contiene 1, ogni uno scritto in IER abiliterà il flag corrispondente in IFR; se esso contiene invece 0, ogni uno scritto nei restanti bit di IER inibirà l'interrupt corrispondente. Facciamo un esempio pratico analizzando la situazione che ci interessa direttamente.

In IER, posto come detto all'indirizzo 37150, si trova memorizzato il numero 130 che corrisponde alla parola binaria 10000010, osserviamo quindi che il bit 2, quello a cui corrisponde il flag di CA1 in IFR, è alto e ciò significa che qualunque cambiamento di stato sulla linea in questione sarà segnalato al sistema. Se noi abbassiamo questo bit memorizzando uno zero, in esso, qualunque transizione su

ABILITAZIONE						
SEI	:	DISABILITA GLI INTERRUPT	:	78	:	120
LDA #B4	:	BYTE BASSO LOCAZIONE DI	:	A9 B4	:	169 180
	:	INIZIO DELLA ROUTINE BEEP	:		:	
STA #0314	:	LO PONE NELLA LOCAZIONE	:	8D 14 03	:	141 20 3
	:	DECIMALE 788	:		:	
LDA #02	:	BYTE ALTO LOCAZIONE DI	:	A9 02	:	169 2
	:	INIZIO DELLA ROUTINE BEEP	:		:	
STA #0315	:	LO PONE NELLA LOCAZIONE	:	8D 15 03	:	141 21 3
	:	DECIMALE 789	:		:	
NOP	:	SPAZIO LIBERO	:	EA	:	234
NOP	:	SPAZIO LIBERO	:	EA	:	234
NOP	:	SPAZIO LIBERO	:	EA	:	234
NOP	:	SPAZIO LIBERO	:	EA	:	234
NOP	:	SPAZIO LIBERO	:	EA	:	234
CLI	:	ABILITA GLI INTERRUPT	:	58	:	88
RTS	:	RETURN	:	60	:	96

ROUTINE RUN/STOP						
JSR #FF9F	:	ROUTINE TASTIERA	:	20 9F FF	:	32 159 255 02B4
LDA #04	:	4 DECIMALE	:	A9 04	:	169 4
BIT #028D	:	ESEQUE L'AND DI A CON IL	:	2C 8D 02	:	44 141 2
	:	CONTENUTO DEL REG. 653	:		:	
BNE 02B4	:	SE NON E' SODDISFATTO	:	D0 F6	:	208 246
	:	SALTA ALLA ROUTINE TAST.	:		:	
JMP #EABF	:	SALTA ALL'INTERRUPT ORIG.	:	4C BF EA	:	76 191 234

Listato 4 - Questa routine può essere usata per arrestare lo scroll del listato basic di un programma dopo il LIST premendo il tasto CTRL.

```

10 REM * RUN/STOP *
20 REM
30 FOR I=674T0704
40 READ A:POKE I,A:NEXT
50 DATA 120,169,180,141,20,3,169,2,141,21
60 DATA 3,234,234,234,234,234,88,96,32,159
70 DATA 255,169,4,44,141,2,208,246,76,191,234

```

Listato 5 - Programma Basic per introdurre in macchina la routine del listato 4.

```

10 REM * BEEP SOFT + OFF RESTORE
20 REM
30 FOR I=674T0741
40 READ A:POKE I,A:NEXT
50 DATA 120,169,185,141,20,3,169,2,141,21,3,169
60 DATA 244,141,10,144,169,2,141,30,145,88,96
70 DATA 169,64,44,197,0,208,35,169,15,141,14,144
80 DATA 169,0,141,27,145,169,0,141,24,145,169,255
90 DATA 141,25,145,169,32,44,29,145,248,251,173,24
99 DATA 145,169,0,141,14,144,76,191,234

```

Listato 6 - Questo programma produce un beep sfruttando l'uscita audio del televisore.

CA1 non influirà sul contenuto in IFR. Per effettuare tale operazione, cioè porre a zero il bit 2 di IER, dovremmo memorizzare in esso una parola che contenga uno 0 nel bit 7 ed un 1 in corrispondenza del flag da inibire. La parola binaria in questione è quindi 00000010 che in decimale corrisponde a 2. Provate infatti a digitare POKE 37150,2

e premere RETURN per constatare che la pressione contemporanea di RUN/STOP e RESTORE non influirà più sul sistema come all'origine.

Nel listato 3 riportiamo la routine in LM che effettua le operazioni suddette. Essa potrà essere sostituita nelle posizioni del

listato 1 occupate dall'istruzione NOP. Per rendere attiva questa sezione nel listato 2 dovrete sostituire i numeri decimali che compaiono nella colonna più a destra del listato 3 al posto delle posizioni occupate dal numero 234 nella linea 50.

### Una comoda utility

Come ultima cosa esaminiamo una semplice routine che serve per arrestare le funzioni della macchina tenendo premuto il tasto CTRL (ConTRoL) ed a riabilitarla con il rilascio. Questa funzione può essere utilizzata ad esempio per bloccare un listato che scorre sullo schermo tenendo premuto il tasto menzionato.

Essa è riportata nel listato 4 mentre nel 5 riportiamo il programma per introdurla in macchina utilizzando il Basic. Ricordiamo che una volta introdotta la routine, dopo il RUN, il programma Basic può anche essere "cancellato" con il comando NEW. La prima parte della routine è come al solito simile alle altre occupandosi semplicemente di predisporre la macchina ad andare ad eseguire ogni sessantesimo di secondo la nostra utility.

Analizziamo alcuni fatti importanti per la comprensione del segmento denominato ROUTINE RUN/STOP.

A partire dalla locazione FF9F è contenuto un puntatore a due byte che indirizza alla locazione esadecimale EB1E (60190 decimale), da cui parte la routine che verifica se è stato premuto un tasto sulla tastiera. Se la risposta è negativa, essa non ha più seguito. In caso affermativo il tasto viene identificato ed il suo codice ASCII è posto in un buffer (buffer di tastiera) situato tra le locazioni esadecimali 0277 e 0280 (631 e 640 decimale).

Viene poi incrementato il contatore che conta i caratteri nel buffer e situato all'indirizzo esadecimale 00C6 (198 decimale). Inoltre nel byte 023 o 197 è trasferito il codice tastiera del tasto che è stato premuto per ultimo; esso in condizione di riposo contiene il numero decimale 64. Questo vale per tutti i tasti (escluso RESTORE) meno SHIFT, CBM e CTRL il cui codice di tastiera è posto nel byte 028D (653 decimale). Ciò che ci interessa conoscere è che in esso, dopo la pressione di CTRL, è contenuto il numero 4.

La nostra utility effettua per prima cosa il salto alla routine di tastiera che controlla la pressione di un eventuale tasto e compie le operazioni sopra descritte. Viene in seguito caricato nell'accumulatore, A, il numero 4 ed eseguito l'AND di A con il contenuto del byte 653 e con tale operazione si verifica se è stato premuto il tasto CTRL. In caso negativo il controllo passa alle normali routine di manipolazione dell'interrupt mentre se il test dà risultato affermativo, il sistema entra in un loop infinito (con l'invio a 02B4) da cui si esce rilasciando il CTRL.

Nel listato 6 riportiamo la routine per la generazione del beep senza l'ausilio del circuito esterno (già pubblicata la volta scorsa) con l'aggiunta del blocco del RESTORE. Il ritardo può essere regolato cambiando l'ultimo valore di DATA (255) della linea 80.

## Vic da zero Programma Victel - MC n° 29 pag. 107

Quando si lavora ad un programma capita di avere sulla scrivania molti listati dello stesso e, al momento di prelevare la copia da consegnare alla tipografia, può succedere di confondere un foglio con un altro. Ringraziando chi ha segnalato l'errore, vi forniamo la versione corretta della linea incriminata del programma Victel relativo al combinatore telefonico per Vic apparso sul numero 20.

```
190 IF A$ = "L" THEN POKE 37147,64:GOSUB 870:GOTO 150
```

Una nota. Lo stop del timer non avviene tramite la verifica di una variabile (S) ma tramite il test del contenuto della locazione 203 che contiene il codice tastiera del tasto che viene premuto (nel nostro caso 41 - linea 730). È questo che rende abbastanza trasparente il motivo dell'arresto del timer dopo la pressione del tasto S.