

CHIPCORDER

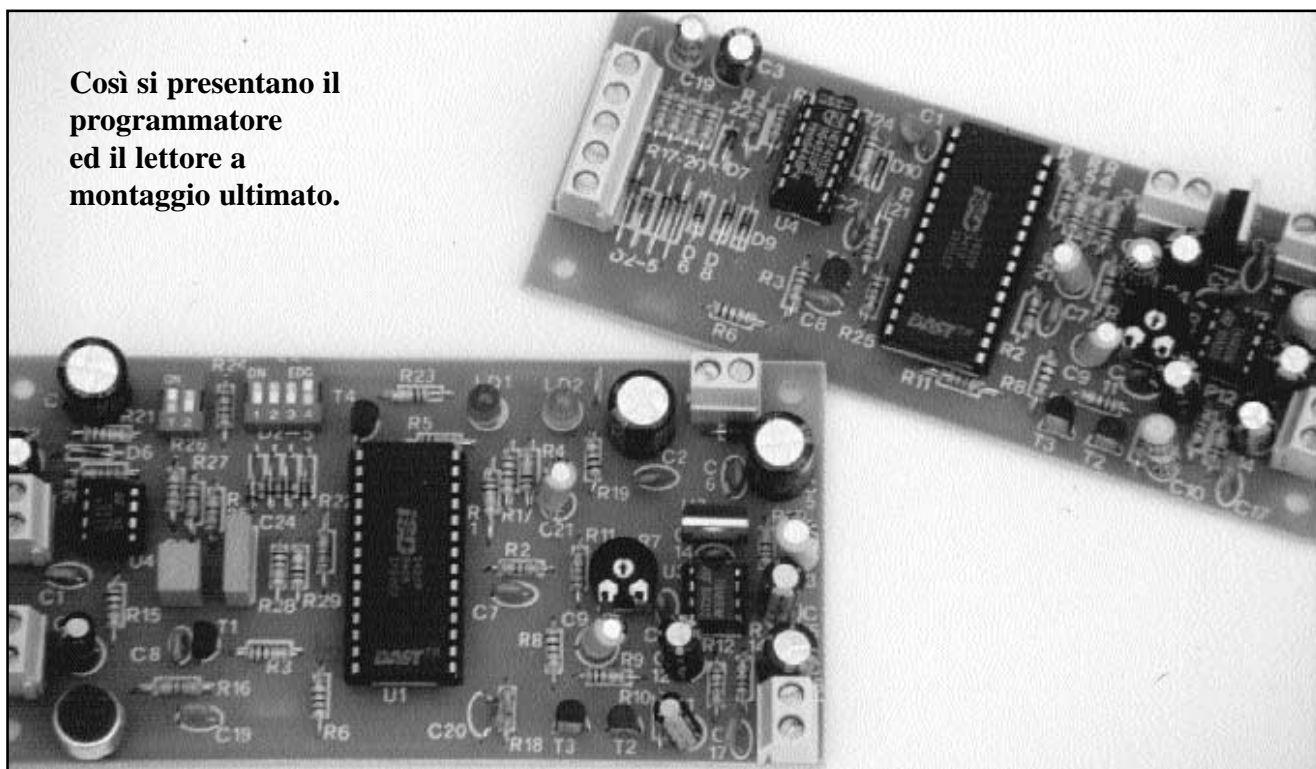
QUATTRO MESSAGGI

di Arsenio Spadoni

L'arrivo della famiglia ChipCorder della ISD, come già a suo tempo per i DAST, ci ha spinto a realizzare una serie di dispositivi per sintesi vocale per applicazioni specifiche; dopo il programmatore ed il lettore a singolo messaggio presentato sul primo numero della nostra rivista, ecco il circuito per registrare sino a quattro messaggi all'interno di un unico chip. Il nuovo programmatore è sostanzialmente simile a quello che si può realizzare con un integrato DAST: cioè permette di registrare in modo permanente (anche se cancellabile elettricamente), da uno a quattro messaggi della stessa

durata, riproducibili dietro un semplice comando. Il programmatore a 4 messaggi che vi proponiamo è insomma un registratore digitale multiplo, simile a quello per il DAST ma semplificato come è possibile solo con i nuovi ChipCorder. Si tratta di un programmatore semplice ma nello stesso tempo molto affidabile, nel quale possiamo notare importanti innovazioni rispetto alla versione con DAST: innanzitutto l'assenza di logica esterna al ChipCorder, che si gestisce da solo le fasi di registrazione e lettura; e poi il temporizzatore di precisione per la durata dei messaggi in registrazio-

Così si presentano il programmatore ed il lettore a montaggio ultimato.



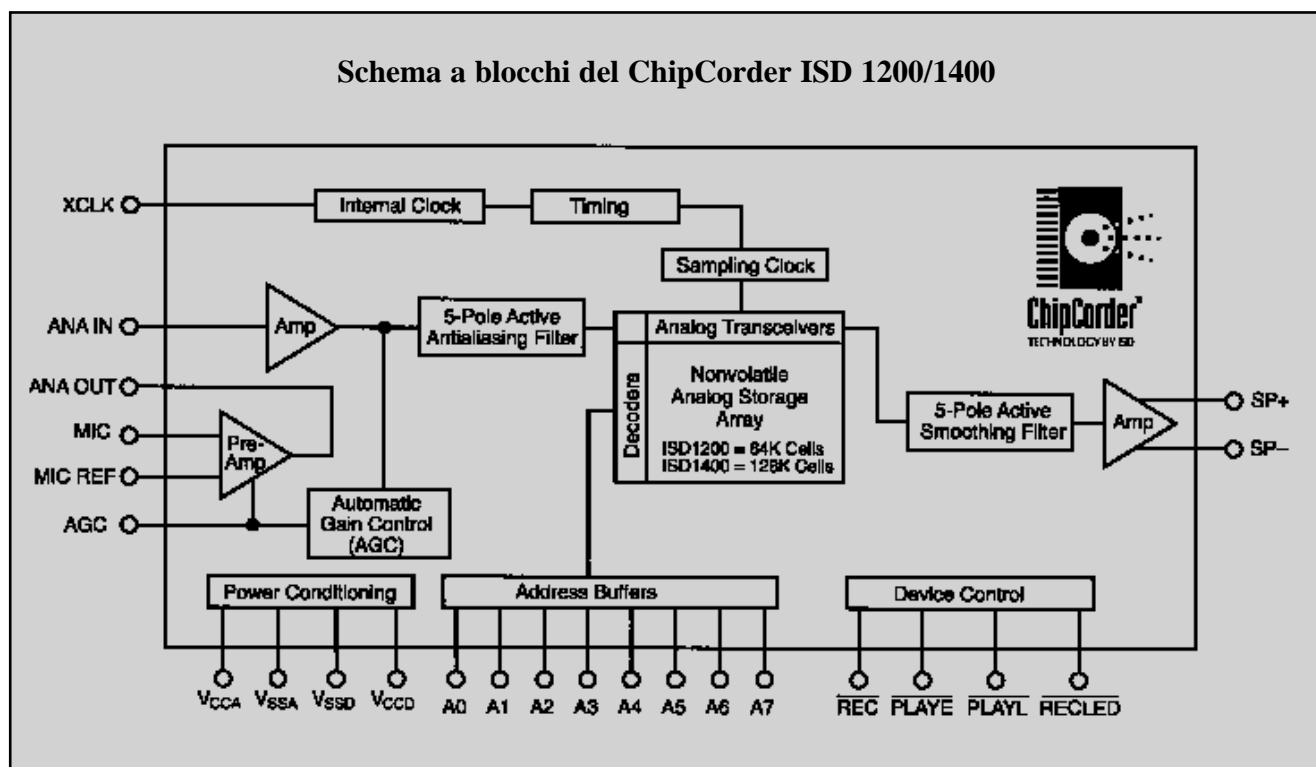
PROGRAMMATORE E LETTORE A DUE O QUATTRO MESSAGGI PER I NUOVI INTEGRATI A SINTESI VOCALE: SUPERDAST, AL SECOLO CHIPCORDER. DUE CIRCUITI SEMPLICI ED AFFIDABILI PER I CHIP VOCALI DEL FUTURO. NATURALMENTE GRAZIE ANCORA ISD!

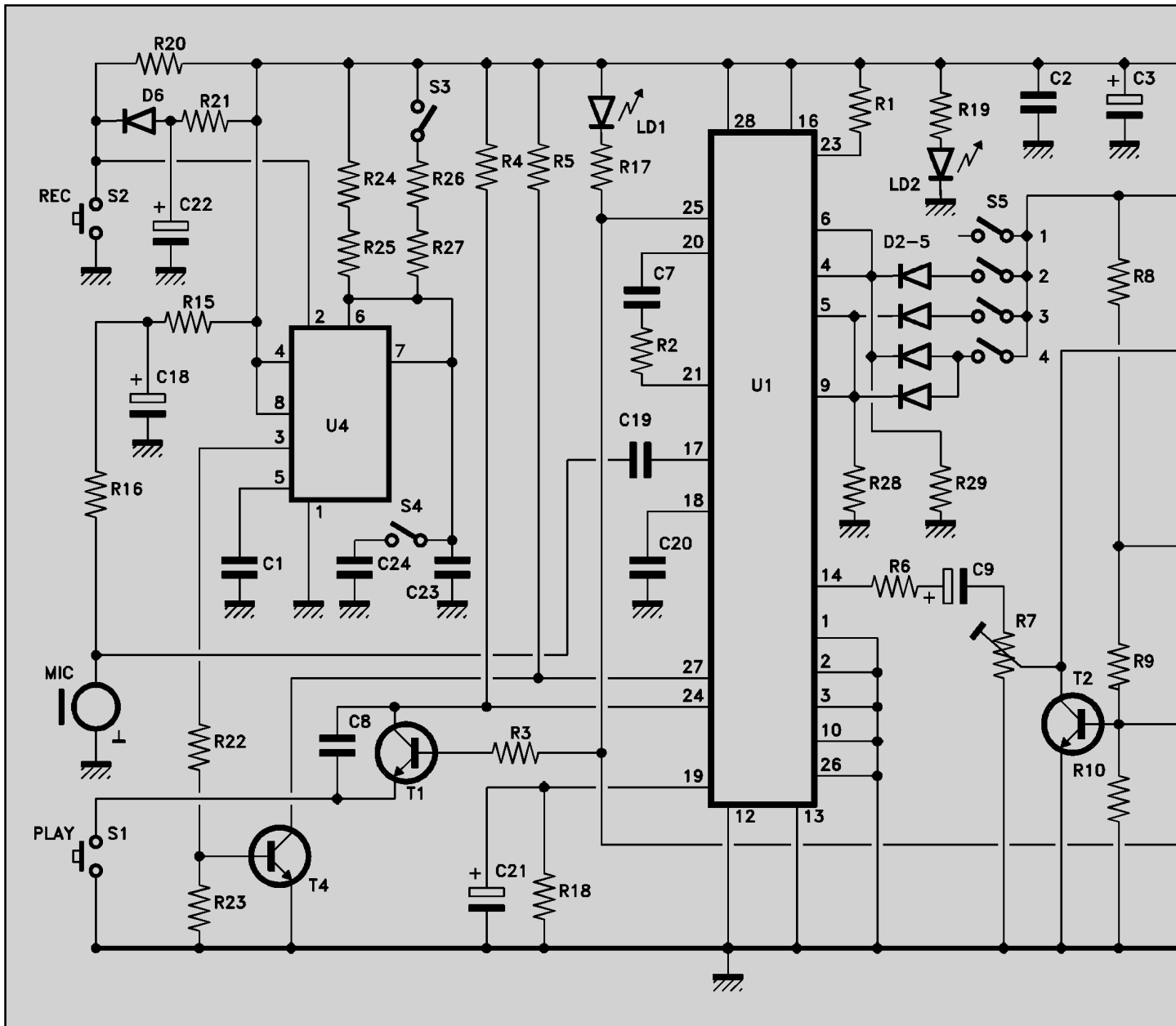


ne, realizzato con un NE555 anziché con le classiche porte logiche NAND. Come potrete constatare, le innovazioni che caratterizzano questo nuovo programmatore esaltano le doti di “maneggevolezza” dei ChipCorder; inoltre l’amplificatore di potenza esterno consente la riproduzione ad un discreto livello sonoro, impossibile da ottenere con il solo finale a ponte incorporato nel ChipCorder. Una buona potenza di uscita del resto non è più un problema, visto che i nuovi integrati ISD assicurano una riproduzione praticamente esente da fruscio di fondo. Di un amplificatore di potenza è

dotato anche il lettore a 4 messaggi che proponiamo insieme al programmatore; il circuito, realizzato anch’esso con un ChipCorder, dispone di tutti gli accorgimenti messi a punto per il programmatore e può riprodurre uno, due, tre, o quattro messaggi registrati distintamente dal programmatore. Quest’ultimo può ovviamente essere utilizzato anche come lettore ma la disponibilità di un lettore specifico è importante in moltissimi casi. Immaginiamo, ad esempio, un impianto vocale per ascensori. Ad ogni piano è installato un lettore che entra in funzione quando arriva l’ascensore;

Schema a blocchi del ChipCorder ISD 1200/1400





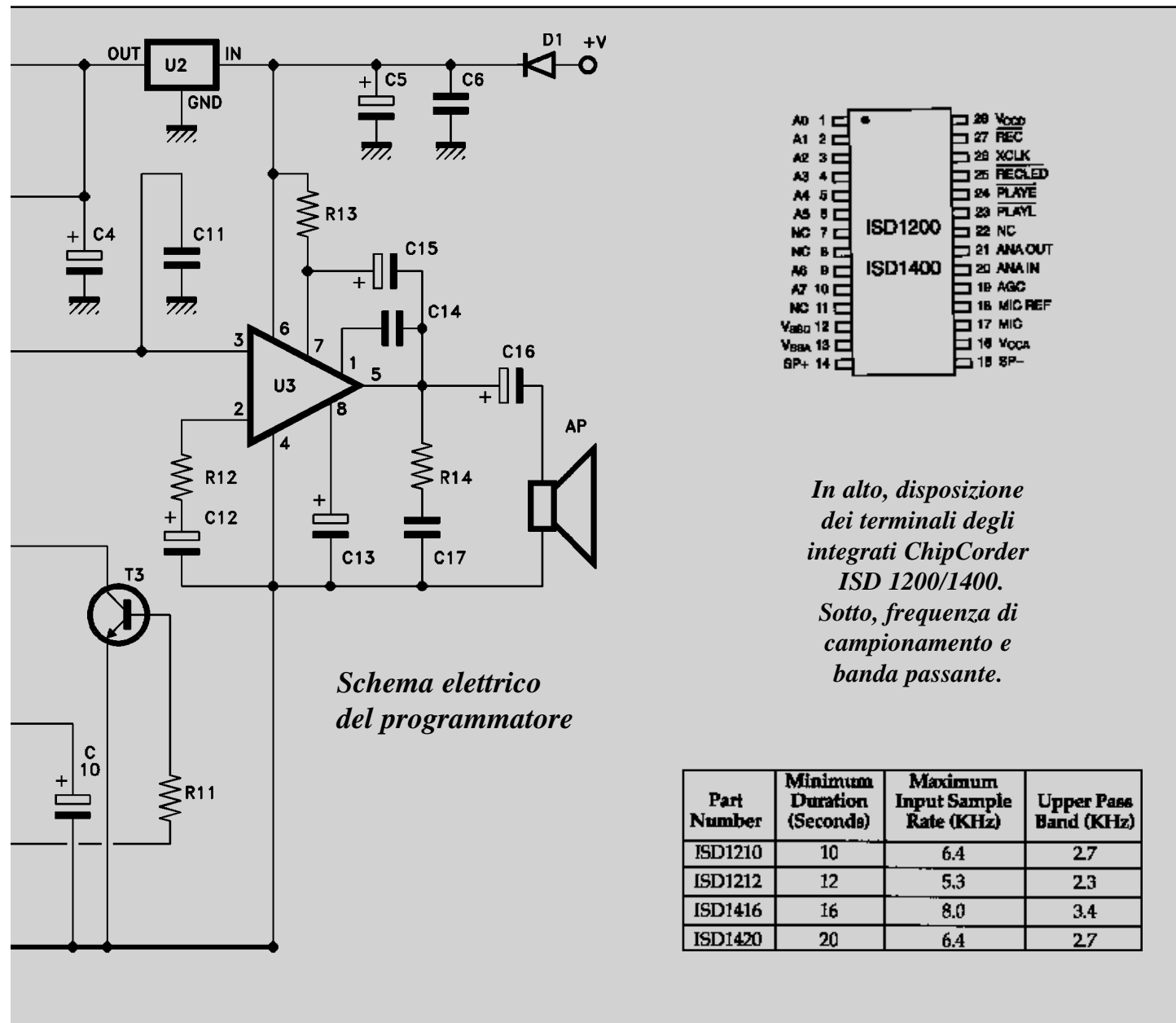
ciascun lettore riprodurrà la frase memorizzata nel ChipCorder, registrata in precedenza col programmatore.

Pertanto per realizzare questo impianto dovremo utilizzare tanti lettori quanti sono i piani, altrettanti integrati ed un

solo programmatore per la memorizzazione delle frasi nei chip. Il programmatore è un condensato di alta tecnologia e dei migliori accorgimenti tecnici adottabili per un circuito di sintesi vocale single-chip. Non stiamo ad arricchire di ulteriori aggettivi il programmatore, perché non ce n'è bisogno, e veniamo al sodo: andiamo a vedere come è fatto questo dispositivo, aiutandoci con lo schema elettrico che trovate in queste pagine. Dunque,



Particolare della scheda di programmazione.

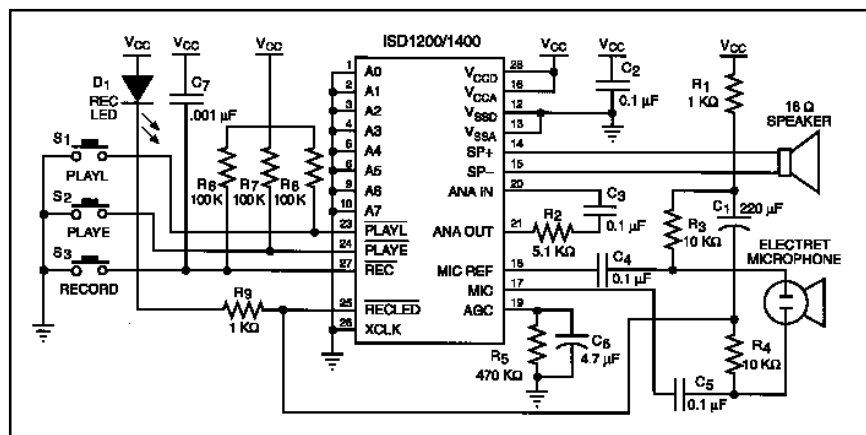


abbiamo detto che il nostro è un programmatore a quattro messaggi; è quindi ovvio che per far entrare tutto in un solo ChipCorder deve poterne ripartire la memoria in quattro parti, quindi deve disporre di appositi comandi per registrare nella prima, nella seconda, nella terza o nella quarta divisione di memoria. Per ottenere ciò abbiamo aggiunto alla configurazione base del programmatore con ChipCorder una serie di switch per impostare i livelli di

Schema applicativo della famiglia ISD 1200/1400.

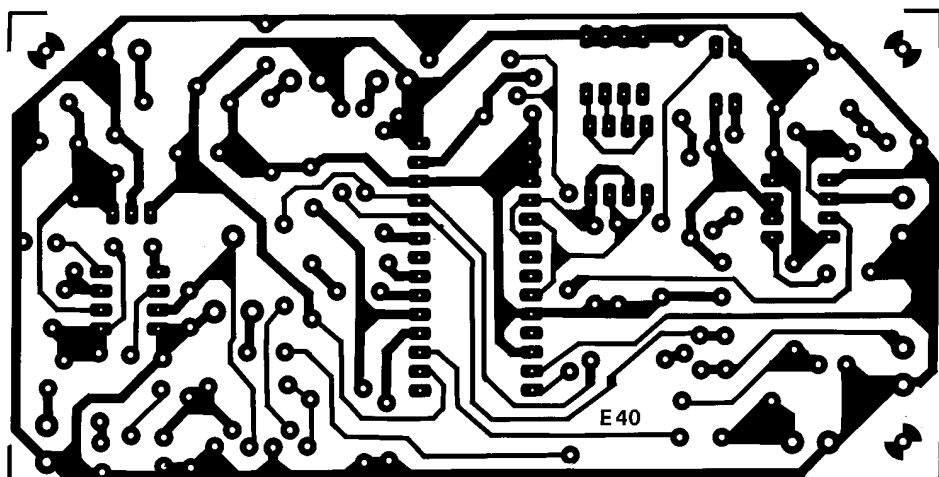
quattro bit d'indirizzamento; in tal modo è possibile decidere da che punto il programmatore deve cominciare a

registrare. In questo caso la registrazione si può avviare con un solo comando indipendentemente dal messaggio che

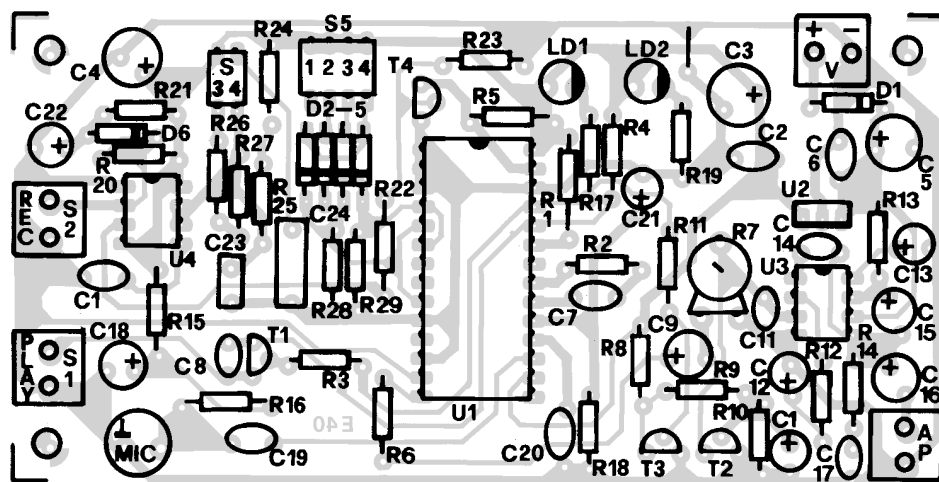


Piano di cablaggio del programmatore a 2/4 messaggi

Circuito stampato in scala 1:1



Disposizione dei componenti



COMPONENTI

- R1: 100 Kohm
- R2: 4,7 Kohm
- R3: 10 Kohm
- R4: 100 Kohm
- R5: 100 Kohm
- R6: 1 Kohm
- R7: 10 Kohm trimmer
- R8: 1 Kohm
- R9: 10 Kohm
- R10: 22 Kohm
- R11: 22 Kohm
- R12: 150 ohm
- R13: 56 ohm
- R14: 1 ohm
- R15: 1 Kohm
- R16: 4,7 Kohm
- R17: 1 Kohm
- R18: 470 Kohm
- R19: 1 Kohm
- R20: 22 Kohm
- R21: 10 Kohm
- R22: 22 Kohm
- R23: 100 Kohm
- R24: 3,3 Mohm
- R25: 3,3 Mohm
- R26: 3,3 Mohm
- R27: 2,7 Mohm
- R28: 10 Kohm
- R29: 10 Kohm
- (I resistori sono da 1/4 W 5 %)
- C1: 100 nF multistrato
- C2: 22 nF ceramico
- C3: 470 µF 16 V elettr. rad.
- C4: 470 µF 16V elettrolitico rad.
- C5: 470 µF 25V elettrolitico rad.
- C6: 22 nF ceramico
- C7: 100 nF multistrato
- C8: 100 nF multistrato
- C9: 1 µF 16V elettrolitico rad.
- C10: 10 µF 16V elettrolitico rad.
- C11: 220 pF ceramico

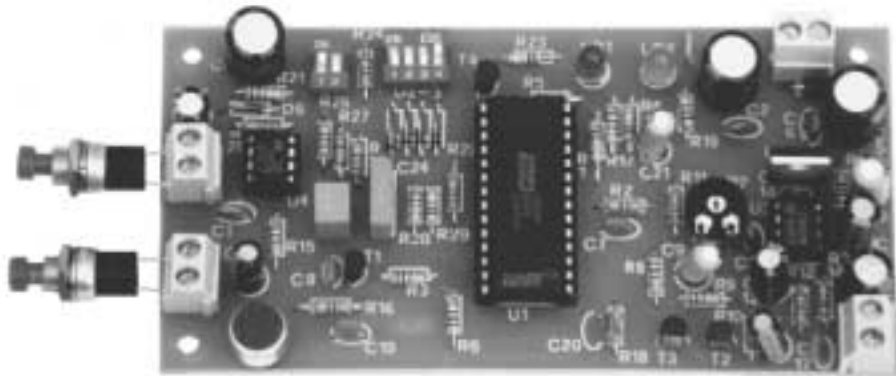
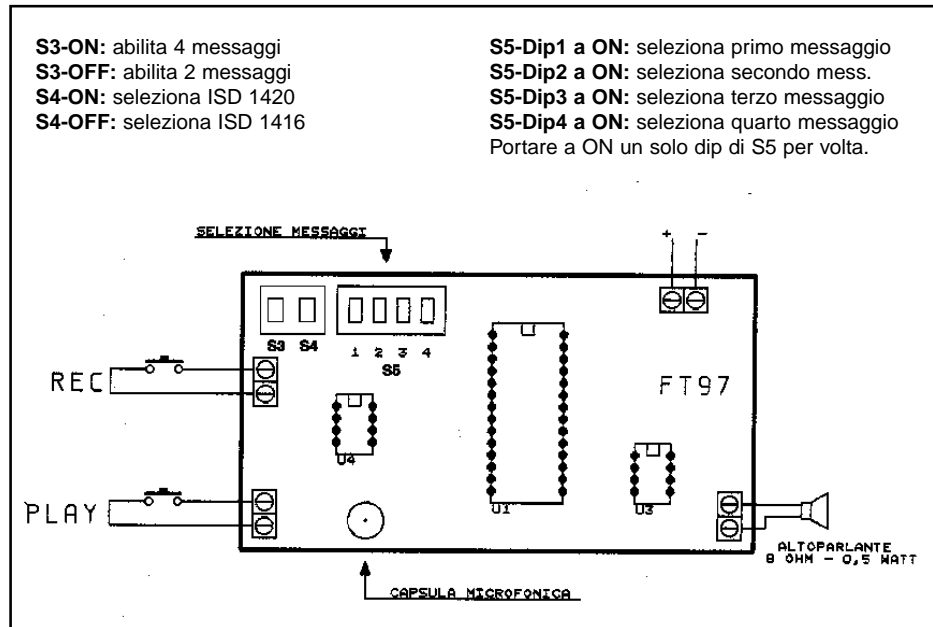
si va a registrare. Risolto il problema della divisione della memoria resta quello dell'arresto del ChipCorder, cioè bisogna far rispettare a ciascun messaggio lo spazio che gli viene assegnato. Sappiamo infatti che registrando, ad esempio, il primo messaggio (prima divisione di memoria) bisogna fermarsi trascorso un quarto del tempo totale a disposizione (in realtà qualcosa meno...) altrimenti si va a scrivere nello spazio riservato al messaggio successivo: il secondo. Quindi come fare per

assicurarsi che non vi siano sconfinamenti? La soluzione più immediata sarebbe registrare con il cronometro, tuttavia si tratta di un metodo scomodo, inaffidabile e poco preciso, perché fermandosi prima si sprecherebbe parte della memoria, mentre arrestandosi oltre il limite si comprometterebbe il tempo a disposizione per il successivo messaggio. Abbiamo quindi pensato alla soluzione ideale: un temporizzatore a cui affidare il ciclo di registrazione per ciascun messaggio. Qualunque sia

la divisione di memoria interessata il comando di programmazione lo dà un temporizzatore, che provvede a revocarlo poco prima dello scadere di un quarto del tempo totale. Così, impiegando un ChipCorder ISD1416 il timer attiva il programmatore per circa 4 secondi a messaggio, mentre usando un ISD1420 la durata di ciascun messaggio è di poco inferiore a 5 secondi. Nel nostro programmatore il timer è un semplice monostabile realizzato col notissimo NE555, uno dei pochi com-

Collegamenti esterni e impostazione dei dip-switch

- C12:** 100 μ F 16V elettr. rad
C13: 47 μ F 16V elettrolitico rad.
C14: 220 pF ceramico
C15: 100 μ F 16V elettr. rad.
C16: 220 μ F 16V elettr. rad.
C17: 100 nF multistrato
C18: 22 μ F 16V elettrolitico rad.
C19: 100 nF multistrato
C20: 100 nF multistrato
C21: 10 μ F 25 V elettr. rad.
C22: 100 μ F 25V elettr. rad.
C23: 1 μ F poliestere
C24: 220 nF poliestere
D1: 1N4002
D2: 1N4148
D3: 1N4148
D4: 1N4148
D5: 1N4148
D6: 1N4148
LD1: LED rosso 5 mm
LD2: LED verde 5 mm
T1: BC547B
T2: BC547B
T3: BC547B
T4: BC547B
U1: ISD1400 da programmare
U2: L7805
U3: TBA820M
U4: 555 (SGS)
AP: Altoparlante 8 ohm 0,5 W
MIC: Capsula microfonica preamplificata a due fili
S1: Pulsante n. a.
S2: Pulsante n.a.
S3-4: Dip-switch a 2 vie
S5: Dip-switch a 4 vie
Varie:
 - Zoccolo 4+4 pin
 - Zoccolo 14+14 pin
 - Morsetto 2 poli passo 5 (4 pz)
 - C.S. cod. E40

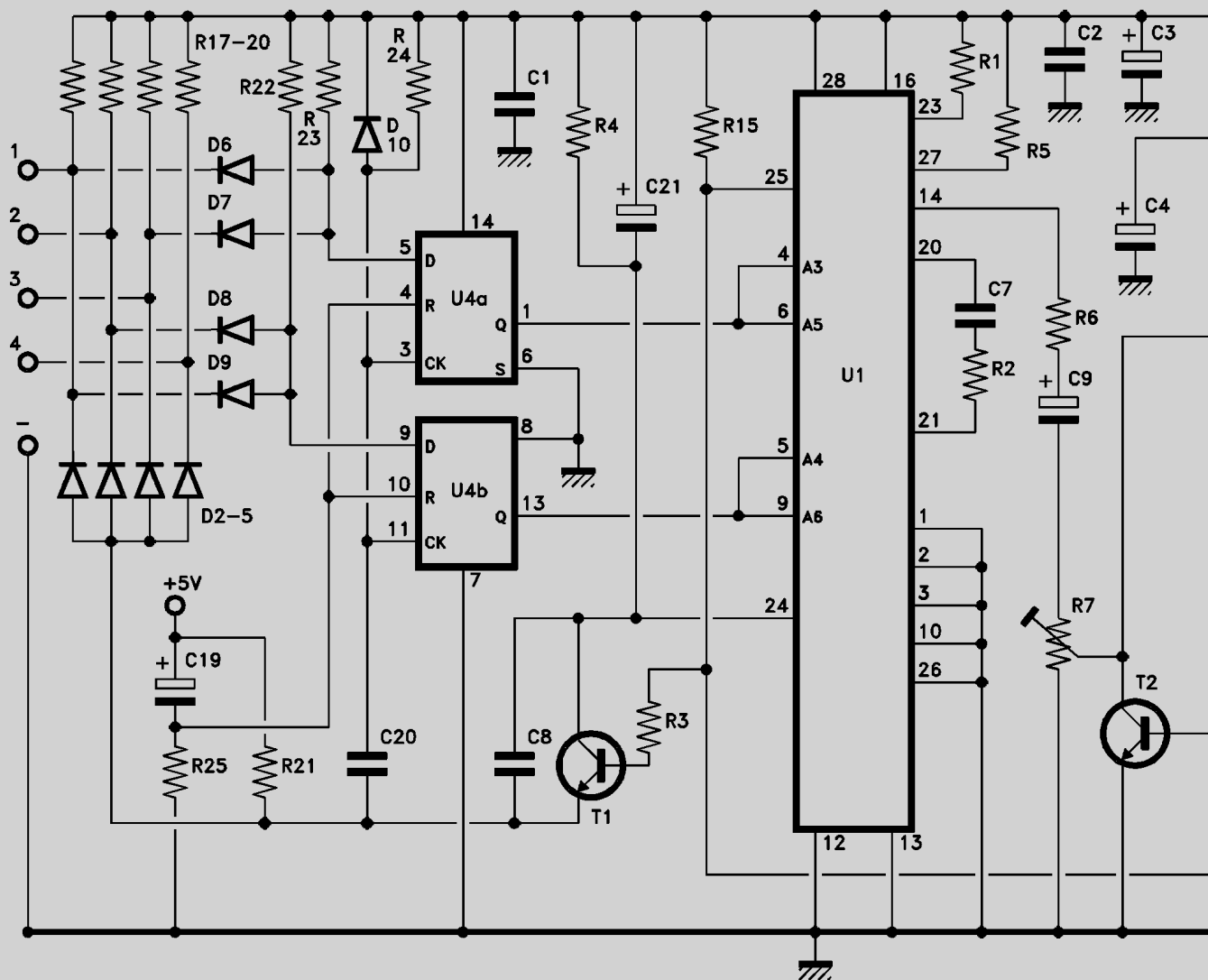


ponenti idonei a realizzare temporizzatori di elevata precisione. Il monostabile ha una rete di temporizzazione con alcuni elementi inseribili per ottenere i tempi relativi all'utilizzo dell'ISD1416 (16 secondi) e dell'ISD1420 (20 secondi) sia nel funzionamento a 2 messaggi che in quello a quattro. Lo switch S4 permette di scegliere il tipo di integrato da usare: deve stare aperto impiegando un ISD1416 mentre montando un ISD1420 deve essere chiuso, in modo da inserire la massima capacità ottenen-

do il maggior tempo di registrazione. Lo switch S3 serve invece per comunicare al timer se si deve registrare a due o a quattro messaggi: interruttore aperto significa registrare a due messaggi (infatti la resistenza di temporizzazione è maggiore e determina il tempo più lungo) mentre interruttore chiuso equivale alla registrazione a quattro messaggi (tempo per messaggio più che dimezzato). Sapendo che l'NE555 funzionando da monostabile dà in uscita impulsi positivi della durata di

$1,1 \times R \times C$, dove R e C sono resistenza e condensatore posti rispettivamente tra i piedini 6,7 e l'alimentazione, e tra i medesimi e massa, possiamo facilmente determinare cosa accade "manovrando" gli switch S3 e S4: lasciandoli entrambi aperti l'impulso del monostabile dura circa 7,3 secondi, che diventano quasi 8,8 con il solo S4 chiuso. Questi sono i tempi corrispondenti al funzionamento a due messaggi rispettivamente per l'ISD1416 e per l'ISD1420. Naturalmente tali tempi

Schema elettrico del lettore di ChipCorder a 2-4 messaggi



COMPONENTI

R1: 100 Kohm
R2: 4,7 Kohm
R3: 10 Kohm

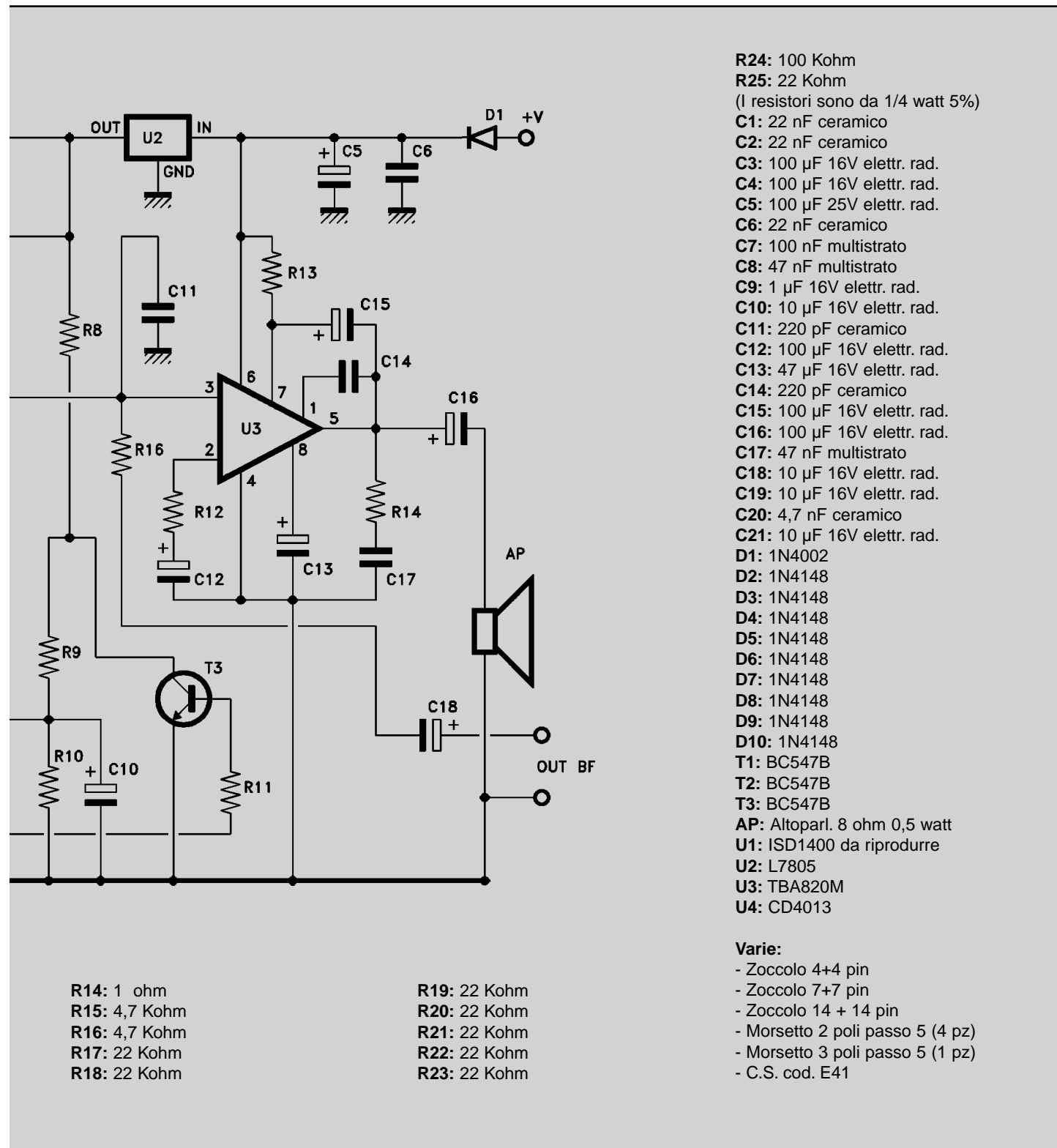
R4: 2,2 Kohm
R5: 100 Kohm
R6: 1 Kohm
R7: 10 Kohm trimmer
R8: 1 Kohm

R9: 10 Kohm
R10: 22 Kohm
R11: 22 Kohm
R12: 150 ohm
R13: 56 ohm

sono inferiori alla metà della capacità dei rispettivi integrati, e lo sono perché viste le tolleranze dei componenti la scelta di tempi troppo vicini ai limiti avrebbe determinato in qualche caso l'invasione del tempo riservato al messaggio seguente. Con tempi sensibilmente più brevi ci mettiamo al riparo da

tale inconveniente, anche se dobbiamo rinunciare ad un filino di tempo effettivamente utilizzabile. Con il solo S3 chiuso si ottengono invece impulsi della durata di circa 3,4 secondi, che corrispondono al funzionamento a 4 messaggi con un ISD1416; per l'ISD1420 a quattro messaggi occorre

chiudere anche S4, allorché il monostabile produce impulsi della durata di circa 4,4 secondi. Quindi, tirando le somme, possiamo dire che per l'ISD1416 abbiamo a disposizione due messaggi da poco più di 7 secondi cadauno o quattro da circa 3 secondi e mezzo, mentre registrando in un

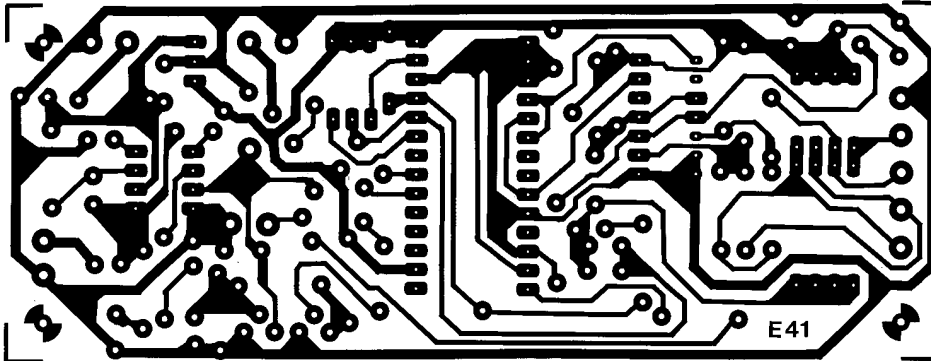


ISD1420 possiamo realizzare due messaggi da circa 9 secondi o quattro da poco meno di 4,5 secondi. Torniamo ora allo schema elettrico del programmatore per vedere come avviene in pratica la registrazione: l'avvio della relativa fase si ottiene triggerando il monostabile U4, ovvero pigiando per un

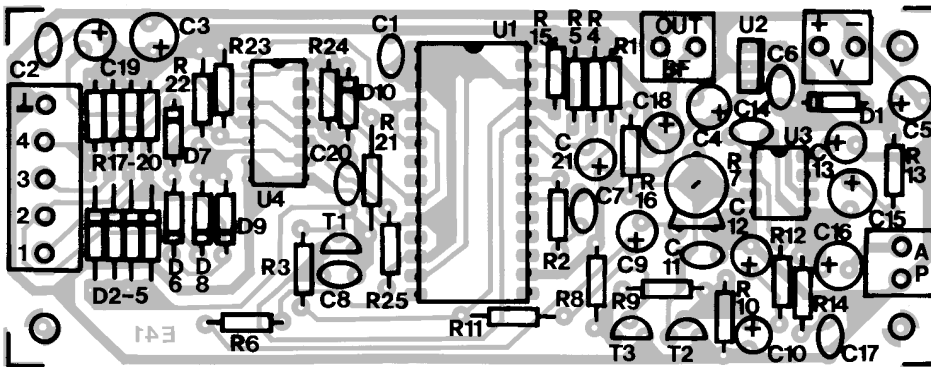
istante il pulsante S2 (REC); così facendo l'NE555 produce tra il proprio piedino 3 e massa un impulso positivo che manda in saturazione T4. Il collettore di quest'ultimo trascina a massa il piedino 27 (REC) del ChipCorder che va così in registrazione. Notate che all'avvio della registrazione la logica

dell'U1 legge lo stato degli indirizzi e parte dalla locazione di memoria indirizzata. Nel nostro caso abbiamo reso manipolabili quattro bit di indirizzo allo scopo di scegliere quattro possibili punti di partenza per la programmazione; gli indirizzi corrispondono a zero (tutti gli switch di S5 aperti, quindi tutti

Lettores a 2/4 messaggi per ChipCorder ISD



Circuito stampato in scala 1:1



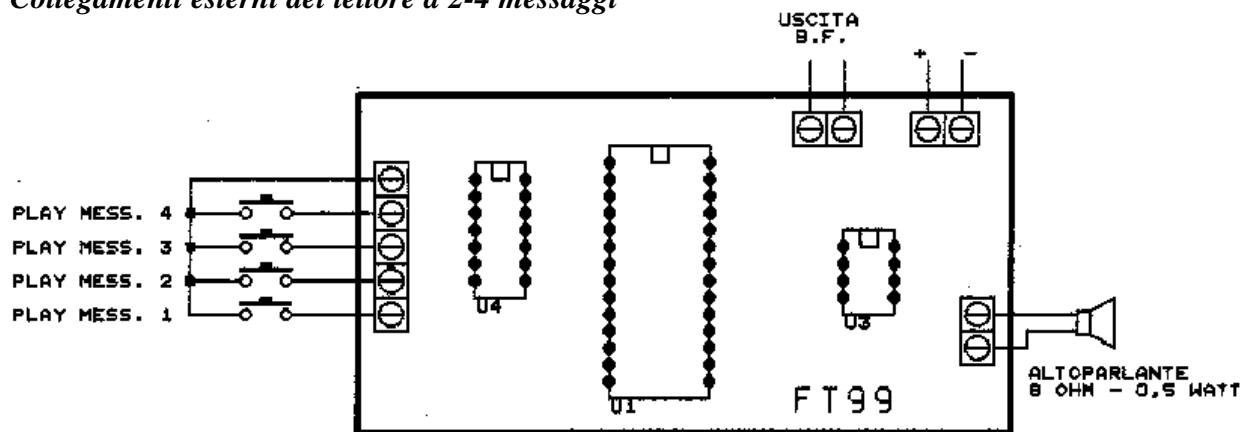
Disposizione dei componenti

gli address a zero logico) per il primo messaggio, a 40 (switch 2 chiuso, ovvero address A5 e A3 a livello alto) per il secondo, a 80 (switch 3 chiuso, quindi address A6 e A4 a livello uno) per il terzo e a 120 (switch 4 chiuso, ovvero A3, A4, A5 e A6 a livello alto) per il

quarto. Poiché la memoria degli ISD1400 è ripartibile in 160 locazioni elementari tutto quadrato: 0-40 è il primo quarto, 40-80 è il secondo, il terzo è tra 80 e 120, mentre 120-160 è lo spazio per il quarto messaggio. Alla luce di ciò è ovvio che per registrare a due mes-

saggi basta indirizzare gli address 0 (per il primo) e 80 (per il secondo) quindi va usato il solo interruttore 3 dell'S5. Qualunque sia la configurazione scelta la registrazione termina quando lo decide il timer, poiché a tempo scaduto il pin 3 dell'NE555 torna a

Collegamenti esterni del lettore a 2-4 messaggi



livello basso e lascia interdire T4, il cui collettore torna a livello alto e lascia nel medesimo stato il pin 27 del ChipCorder. Notate che durante la registrazione resta acceso il LED LD1; notate anche che a fine registrazione, cioè durante il passaggio 0/1 dello stato del pin 27, U1 segna il fine messaggio in memoria. Durante la fase di registrazione l'amplificatore di potenza viene tacitato mediante T2 e T3: infatti in programmazione il pin 25 assume lo zero logico, T3 resta interdetto, e T2 viene saturato e va a cortocircuitare l'uscita del trimmer di regolazione del volume. In tal modo l'altoparlante non può riprodurre suoni e rumori casuali prodotti dal ChipCorder quando registra.

IL RIASCOLTO

Con il programmatore si può anche ascoltare il risultato di ogni registrazione e comunque si possono leggere integrati ChipCorder già programmati: basta premere il pulsante S1, e il chip vocale inizia la lettura della propria memoria. Notate che anche in lettura gli indirizzi decidono da dove il chip deve iniziare a leggere; quindi se volete ascoltare il primo messaggio dovete lasciare aperti gli switch di S5, per il secondo deve rimanere chiuso il solo switch 2, per il terzo messaggio va posto a ON il terzo dip ed infine per ascoltare il quarto messaggio va posto a ON l'ultimo dip. La riproduzione si arresta automaticamente a fine messaggio, pertanto è evidente che se, ad esempio, ascoltate la seconda parte

della memoria (dall'indirizzo 80 in poi) di un ChipCorder registrato a 4 messaggi, chiudendo il solo switch 3 dell'S5, non potete che ascoltare il solo terzo messaggio; il quarto non potete leggerlo perché alla fine del terzo il ChipCorder rileva il fine-messaggio e si blocca automaticamente. Perciò ogni integrato va "ascoltato" nello stesso modo in cui è stato registrato: a due messaggi se è stato ripartito in due, a quattro messaggi se è stato suddiviso in quattro banchi. In lettura l'amplificatore di potenza esterno riceve il segnale dal piedino 14 dell'U1 (ponte interno al ChipCorder) e lo amplifica rendendolo udibile in altoparlante; infatti in fase di lettura il piedino 25 del ChipCorder sta a livello alto e fa saturare T3, quindi T2 può rimanere interdetto e non disturba il segnale che arriva dal trimmer R7 (controllo di volume).

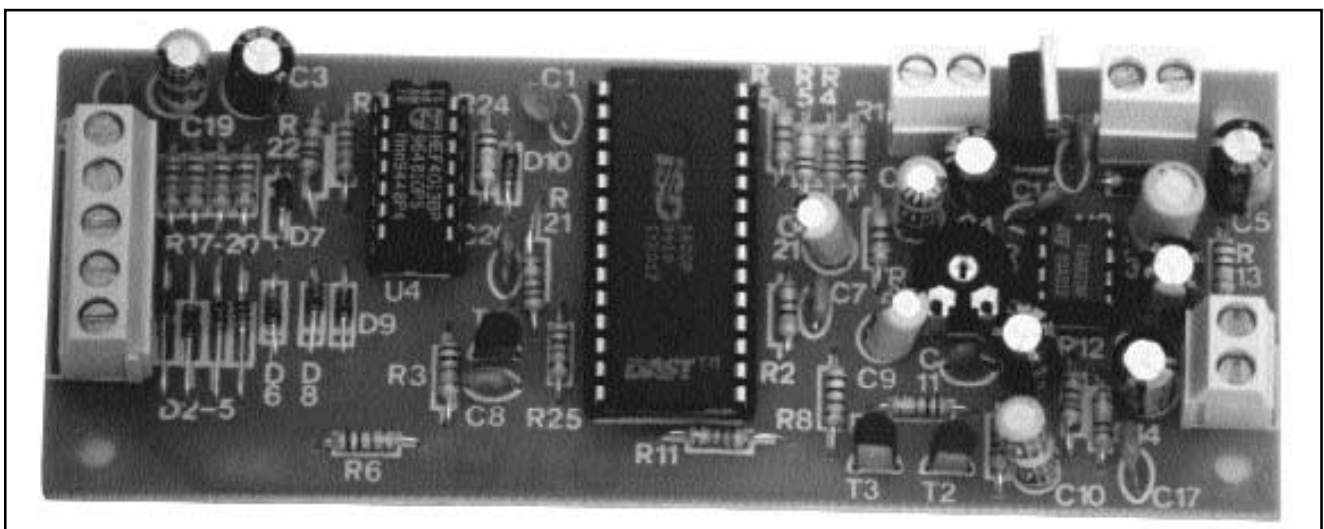
IL RIPRODUTTORE

Bene, con il programmatore abbiamo concluso, o almeno ne abbiamo visto gli aspetti principali; cerchiamo ora di fare lo stesso con il lettore a 4 messaggi, del quale analizziamo lo schema elettrico, illustrato in queste pagine. Questo particolare lettore è stato studiato appositamente per riprodurre quanto registrato in quattro parti uguali della memoria di un ChipCorder; l'assenza di temporizzatori e la conoscenza del funzionamento di tale integrato vocale ci permette di dire con certezza che il lettore ignora il tipo di ChipCorder che gli viene montato.

Infatti dispone solo della logica necessaria ad impostare gli indirizzi di partenza dei quattro messaggi di un integrato serie 1400 (a 160 partizioni di memoria); gli indirizzi sono, lo ricordiamo, zero, 40, 80, 120. E' quindi ovvio che non si può leggere a quattro messaggi un chip ISD1200, che peraltro non è neppure programmabile col dispositivo che proponiamo. Allora, il funzionamento del lettore è molto semplice: ci sono quattro ingressi per l'abilitazione di ciascuno dei 4 messaggi; ogni ingresso è attivo a livello basso ed una volta eccitato determina due situazioni: fa impostare gli indirizzi relativi al proprio messaggio, quindi attiva il ChipCorder. Vediamo come avviene tutto ciò, considerando l'ingresso relativo al primo messaggio.

GLI INDIRIZZI

Parliamo del punto 1; portandolo a massa (livello logico basso) si pone a zero logico il pin 5 del flip-flop U4a (tipo D configurato come latch) mediante D6 e D9 si pone a zero anche il Data (pin 9) dell'altro flip-flop: U4b. Tramite D2 si pongono a livello basso gli ingressi di clock dei flip-flop (per mezzo di C20) e l'emettitore del T1; liberando l'ingresso 1 dal circuito, ovvero lasciandolo andare a livello alto (ma anche lasciandolo a zero...) torna ad uno logico la linea di clock ed i flip-flop portano avanti, verso gli address del ChipCorder, gli stati delle loro uscite: zero e zero. Contemporaneamente il chip vocale parte in riproduzione, perché l'impulso dato precedentemente



CON QUALI CHIPCORDER

Il programmatore a quattro messaggi è stato progettato per lavorare con integrati ChipCorder della serie 1400, quindi ISD1416 (da 16 secondi) e ISD1420 (20 secondi); non può quindi programmare correttamente gli ISD1200 (10 e 12 secondi). Lo stesso dicasi per il lettore. Il tutto dipende dalla suddivisione della memoria EEPROM, che nella serie ISD1200 consta di 80 partizioni (64 mila celle) che sono invece il doppio (160, equivalenti a 128 mila celle) negli ISD1400.

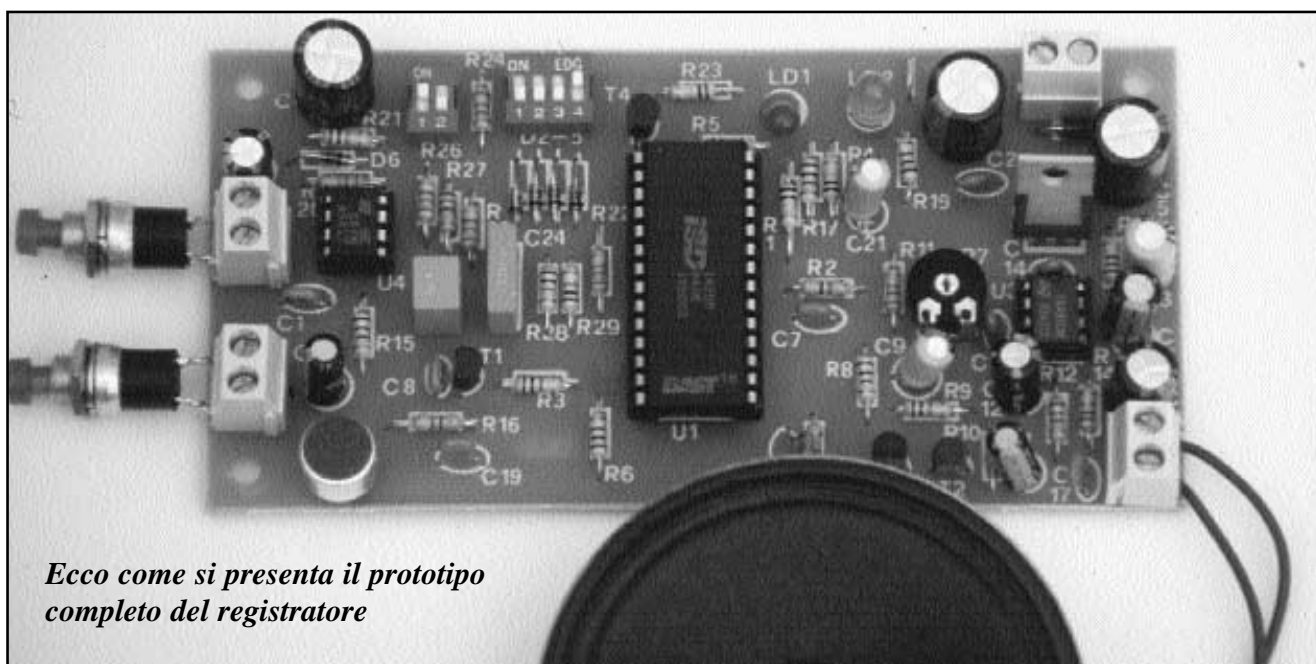
CON IL TEXTTOOL

Volendo utilizzare il programmatore per eseguire una gran quantità di registrazioni su integrati diversi conviene montare uno zoccolo Textool da 28 pin invece di quello tradizionale.

attraverso C8 ha attivato il suo ingresso ad impulso (PLAYE) per la lettura; l'U1 riproduce il contenuto della propria memoria a partire dall'inizio. Si ferma automaticamente al termine del messaggio (allorché riconosce l'EOM); per tutta la fase di lettura l'amplificatore facente capo ad U3 rende udibile in altoparlante il segnale uscente dal ChipCorder. Vediamo ora cosa accade attivando il secondo ingresso: D8 porta a zero logico il piedino Data del flip-flop U4b mentre D3 porta a zero C20 e T1; C20 si ricarica rapidamente (si scaricherà poi attraverso R21, R24 e D10) e i flip-flop ricevono un impulso di clock. Ora l'uscita dell'U4b continua a presentare lo zero logico mentre quella dell'U4a, in virtù dell'uno logico al piedino 5, assume il livello alto. Il ChipCorder va in riproduzione come al

solito, ma inizia a leggere dalla locazione 40: infatti U4a pone a livello alto gli indirizzi A3 (2 alla terza, cioè 8) e A5 (2 alla quinta, cioè 32) che per somma danno appunto 40. Viene riprodotto il secondo messaggio. Attivando l'ingresso 3 viene invece riprodotto il terzo messaggio, poiché mediante D7 viene messo a massa il piedino 5 dell'U4a mentre il 9 dell'U4b resta a livello alto; D4 determina l'eccitazione del clock dei flip-flop quindi il trasferimento alle rispettive uscite degli stati ai piedini Data. Mentre U4a porta lo zero, U4b pone a livello alto i piedini 5 e 9 del ChipCorder, che quindi va in lettura riproducendo il terzo messaggio: parte infatti dalla partizione 80, poiché la somma binaria di A4 (2 alla quarta, cioè 16) e A6 (2 alla sesta, cioè 64) è appunto 80. Il quarto messaggio si

ottiene ovviamente attivando l'ingresso 4; volete sapere perché? Semplice: ponendo a massa il punto 4 viene solo messa a massa la linea di clock e quella di attivazione dell'U1; il ChipCorder parte in lettura dalla locazione 120, poiché l'impulso di clock conseguente all'eccitazione del punto 4 fa avanzare alle uscite dei flip-flop due stati uno (entrambi hanno gli ingressi Data a livello alto, dato che il punto 4 non è in alcun modo collegato ad essi). La somma degli address A3-A5 e A4-A6 è infatti 120 (40+80). Notate che per qualunque messaggio riprodotto la fase di lettura si arresta automaticamente a fine messaggio, quando il ChipCorder riconosce l'EOM. Notate anche che l'amplificatore di uscita (U3) viene tacitato nei momenti in cui il ChipCorder si trova a riposo. L'intero lettore a quattro messaggi è alimentato a 5 volt stabilizzati dal regolatore U2, che ricava tale tensione da quella principale che serve a far funzionare l'amplificatore audio di potenza. La presenza di un'uscita supplementare BF ad alta impedenza (OUT BF) permette di prelevare il segnale uscente dal ChipCorder (beneficia del controllo di volume già utilizzato dall'amplificatore U3) in modo da inviarlo, all'occorrenza, ad altri registratori, mixer, amplificatori di potenza maggiore, eccetera. Abbiamo così terminato anche con il lettore; almeno per ciò che riguarda la teoria. Chi vuol passare ai



Ecco come si presenta il prototipo completo del registratore

fatti continui a leggere perché vedremo i principali aspetti della realizzazione dei circuiti.

REALIZZAZIONE PRATICA

Diciamo prima di tutto che non c'è niente di speciale in alcuno dei due: si tratta di montaggi molto semplici e lo saranno ancor di più seguendo alcune semplici regole come, ad esempio, montare per primi componenti a basso profilo cioè resistenze, diodi e zoccoli, quindi gli altri in ordine di altezza. Facciamo notare che per i condensatori da 47 e 100 nanofarad è bene utilizzare elementi multistrato; per gli altri va bene il normale ceramico a disco. Diodi (sia al silicio che LED) e condensatori elettrolitici hanno una polarità che va rispettata se volete che il circuito funzioni; i transistor ed il regolatore di tensione vanno orientati nel modo indicato nei piani di montaggio illustrati (separatamente per lettore e programmatore) in queste pagine. Sempre in queste pagine trovate le tracce per la realizzazione dei circuiti stampati di programmatore e lettore; per entrambi consigliamo la preparazione mediante la fotoincisione. Per il programmatore va fatto un discorso particolare circa lo zoccolo del ChipCorder: se pensate di impiegare il circuito per programmare spesso diversi integrati consigliamo di usare uno zoccolo Textool, l'unico che non si

DA 16 O 20 SECONDI?

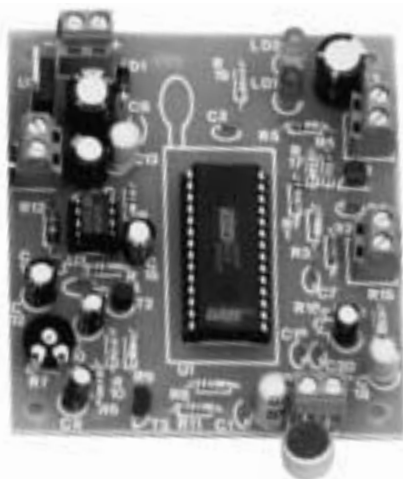
Il programmatore può registrare in integrati ISD1416 o 1420, tuttavia per poterlo fare al meglio deve conoscere preventivamente le caratteristiche del chip che gli viene "affidato". Per questo motivo bisogna settare opportunamente gli interruttori del dip-switch S4. Per programmare un integrato ISD1416 (16 secondi) occorre lasciare aperto S4, che va invece chiuso nel caso di programmazione di un ISD1420 (20 secondi). Ovviamente il programmatore deve conoscere anche in quante parti deve suddividere la memoria del ChipCorder: due o quattro (corrispondenti ad altrettanti messaggi); anche in questo caso si agisce su un dip switch, precisamente su S3. Tale interruttore va lasciato aperto dovendo registrare a due messaggi, mentre va chiuso volendo programmare quattro messaggi.

deteriora con l'uso e che consente di inserire ed estrarre gli integrati senza fatica e con la certezza di buoni contatti elettrici. Per il programmatore facciamo presente che microfono, altoparlante e pulsanti vanno all'esterno dello stampato; la capsula microfonica va connessa secondo la polarità indicata nello schema elettrico, ricordando che il terminale negativo è quello collegato alla carcassa metallica. Quanto al lettore, anche in questo caso l'altoparlante va collegato all'esterno, mediante fili. Per i punti 1, 2, 3, 4, consigliamo di montare dei morsetti per circuito stampato a passo 5 mm, a cui potrete collegare dei pulsanti o fili per l'attivazione dei messaggi.

Entrambi i circuiti vanno alimentati con una tensione continua, meglio se stabilizzata, di valore compreso tra 12 e 15

volt; la corrente richiesta per entrambi è di almeno 350 milliampère.

ANCHE A SINGOLO MESSAGGIO



Sono disponibili anche le schede di programmazioni e di riproduzione a singolo messaggio (memoria intera). Questi dispositivi sono stati descritti nello scorso numero della rivista. Il programmatore (a sinistra) consente la registrazione e il riascolto mediante due soli pulsanti: play e rec. Il riproduttore (a destra) è completo di altoparlante e di amplificatore da 1 watt. Entrambi i circuiti sono disponibili in scatola di montaggio.

