

il telefono sotto controllo



La telefonia digitale sta entrando in tutte le abitazioni con interfacce ISDN a due linee. Vediamo un modo per tenerle tutte sotto controllo

Nel mese di Novembre dello scorso anno, abbiamo presentato un mini centralino telefonico gestito da un computer che riusciva a monitorare, in termini di chiamate uscenti, una linea telefonica standard, registrando numeri

fatti e tempo di chiamata. Ci sono giunte però diverse segnalazioni di lettori che, pur non avendo particolari attività di tipo commerciale, ci chiedevano come riuscire ad espandere tale interfaccia per poter monitorare più linee telefoniche

con la medesima porta seriale, in virtù del fatto che le moderne linee ISDN dispongono di ben 2 canali per la fonia analogica e digitale. Con la soluzione precedente, infatti, ciò sarebbe stato possibile solamente realizzando due circuiti identici e collegandoli ciascuno ad una porta seriale del proprio personal computer.

Tuttavia questa soluzione sarebbe stata praticabile solo da coloro che hanno a disposizione sul computer due seriali disponibili, quindi praticamente da pochissimi, anche se ad oggi tutti i nuovi Pentium II dispongono di mouse con attacco PS/2 che consente di lasciar libere le due seriali.

Poiché sempre più spesso troviamo linee ISDN nelle comuni abitazioni, anche perché il costo di abbonamento non è molto alto e le prestazioni ottenute sono notevoli, abbiamo deciso di offrire un circuito più completo del precedente, che riuscisse a gestire in contemporanea fino ad un massimo di 4 linee telefoniche

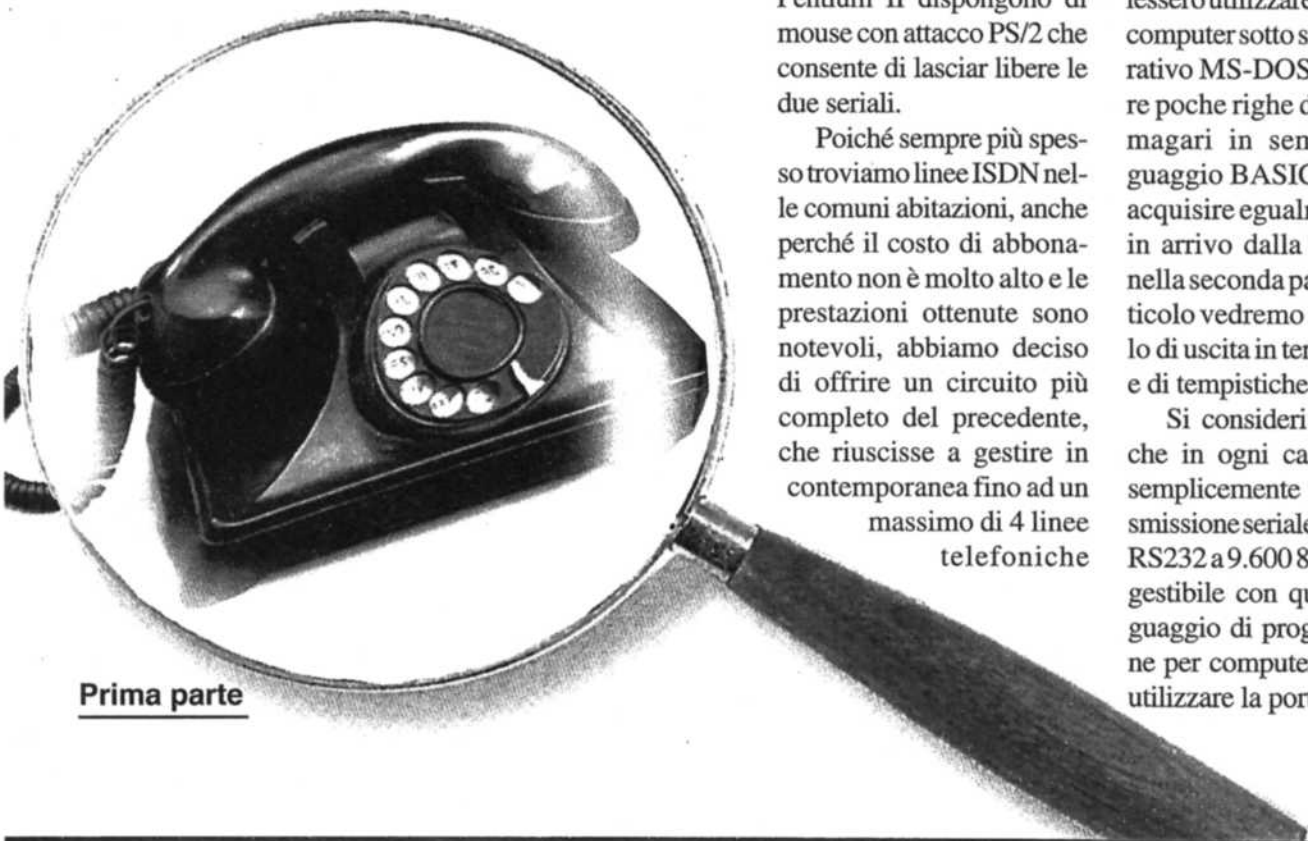
sulla medesima porta seriale sempre di tipo RS232.

Come con l'interfaccia precedente, tutte le operazioni di memorizzazione dei dati vengono affidate ad un computer, gestito da un apposito software che consente all'utente facili operazioni di lettura e di stampa dei dati acquisiti.

Anche in questo caso si è scelto il linguaggio Visual Basic 5 e quindi tale software girerà solamente su computer con sistema operativo Windows 95 o Windows 98.

Resta comunque la possibilità, per coloro che volessero utilizzare un vecchio computer sotto sistema operativo MS-DOS, di scrivere poche righe di software, magari in semplice linguaggio BASIC, per poter acquisire egualmente i dati in arrivo dalla centralina: nella seconda parte dell'articolo vedremo il protocollo di uscita in termini di dati e di tempistiche.

Si consideri comunque che in ogni caso si tratta semplicemente di una trasmissione seriale in standard RS232 a 9.600 8N1 e quindi gestibile con qualsiasi linguaggio di programmazione per computer che possa utilizzare la porta seriale.



Prima parte

La nostra proposta in generale

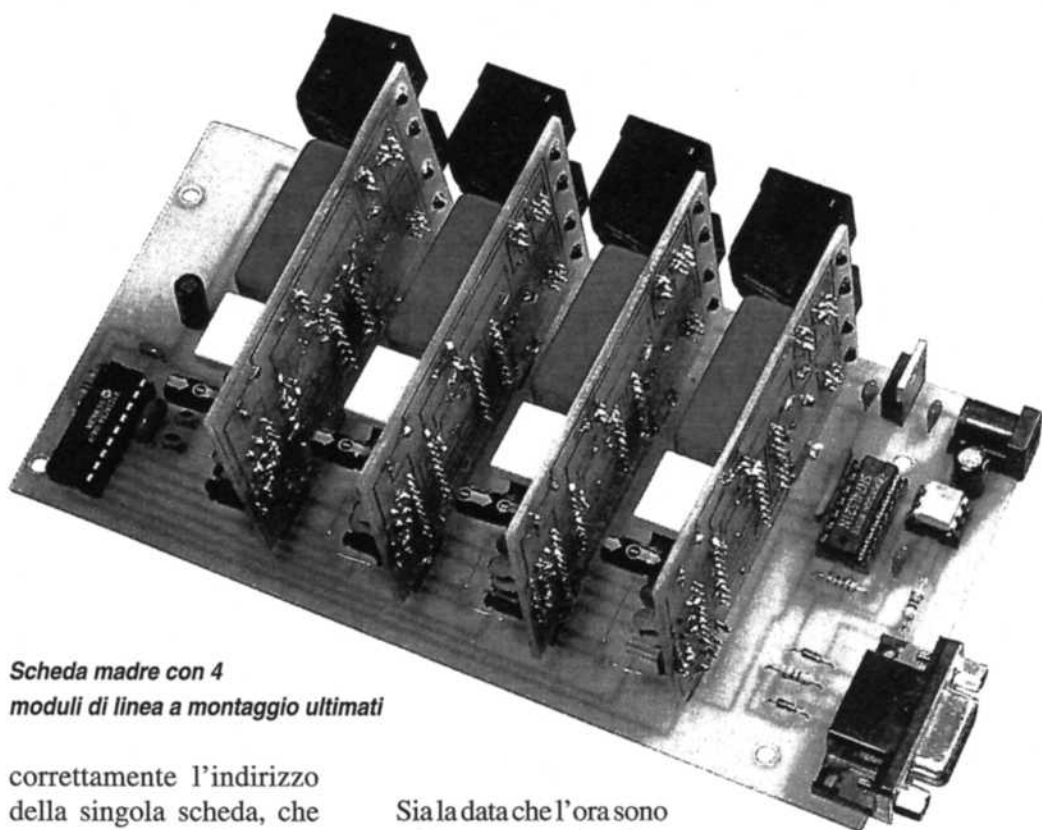
Prima di vedere materialmente il circuito dell'interfaccia, desideriamo mostrare quali siano le caratteristiche elettriche e di funzionamento del sistema, prendendolo in esame nella sua globalità. Il sistema si compone di una scheda "madre" su cui possono essere inserite da 1 a 4 piccole schede "di acquisizione".

Ogni scheda di acquisizione gestisce una linea telefonica indipendentemente dalle altre. Se quindi ad oggi avete la necessità di monitorare una sola linea telefonica ma pensate nel futuro di aggiungerne un'altra, con questo sistema sarà possibile semplicemente inserendo una scheda identica alla prima.

In questo modo, potrete arrivare a controllare fino a 4 linee telefoniche impegnando una sola porta seriale RS232 del computer.

Da prestare attenzione al fatto che il sistema funziona sempre, indipendentemente dal numero di linee prescelto, e che l'inserimento o la rimozione di una scheda di acquisizione non comporta assolutamente malfunzionamenti di alcun genere.

Il software da noi preparato, infatti, riesce a riconoscere le informazioni che gli giungono dalla scheda madre e le organizza in funzione della scheda che le ha inviate: in altri termini, non sarà mai possibile, una volta settato



Scheda madre con 4 moduli di linea a montaggio ultimati

correttamente l'indirizzo della singola scheda, che venga assegnato ad una certa linea telefonica un numero selezionato su di un'altra linea.

Il software preparato, visualizza le informazioni ricevute su quattro finestre, ognuna relativa alla propria linea telefonica.

Contemporaneamente viene creato un file che sarà poi possibile stampare o rivisualizzare in futuro in cui verranno inserite tutte le informazioni relative alle telefonate effettuate.

Ma quali sono queste "informazioni" più volte citate? Si consideri che ogni riga visualizzata o comunque memorizzata, avrà sempre la stessa sintassi, ovvero il primo numero che troveremo identificherà la linea interessata (da 0 a 3).

Il secondo campo informativo sarà la data di inizio della telefonata mentre il terzo campo l'ora di inizio.

Sia la data che l'ora sono complete, ovvero la data comprende l'anno, il mese ed il giorno mentre l'ora comprende le ore, i minuti ed i secondi. Il quarto campo informativo è il numero di telefono composto ed infine il quinto campo sarà l'ora completa relativa alla fine della telefonata.

Il circuito elettrico

Passiamo quindi all'analisi del circuito elettrico, visibile in figura 1.

Si distinguono subito due sezioni: la prima è quella di interfaccia con la linea telefonica, la seconda è quella di controllo dei dati e di dialogo con la scheda madre.

La sezione di interfaccia è a sua volta divisa in due blocchi: uno si occupa di monitorare l'apertura della linea ed i numeri formati con il sistema decadico,

l'altro di acquisire i numeri formati con la codifica DTMF (Dual Tone Mutly Frequency). Tutti ormai conoscono il sistema di codifica DTMF, ma vediamo ugualmente quali sono le caratteristiche principali: tale sistema prevede l'invio di "caratteri DTMF" (sedici in tutto) con cui codificare tutti i numeri da "0" a "9" più altri sei caratteri alfanumerici che sono * # A B C D E F.

Ogni carattere viene inviato e ricevuto come combinazione di due frequenze comprese tra 0 e 3 kHz, ovvero nella banda telefonica.

La composizione di queste due frequenze viene effettuata a matrice, cioè ad ogni carattere corrisponde univocamente una frequenza di riga ed una di colonna.

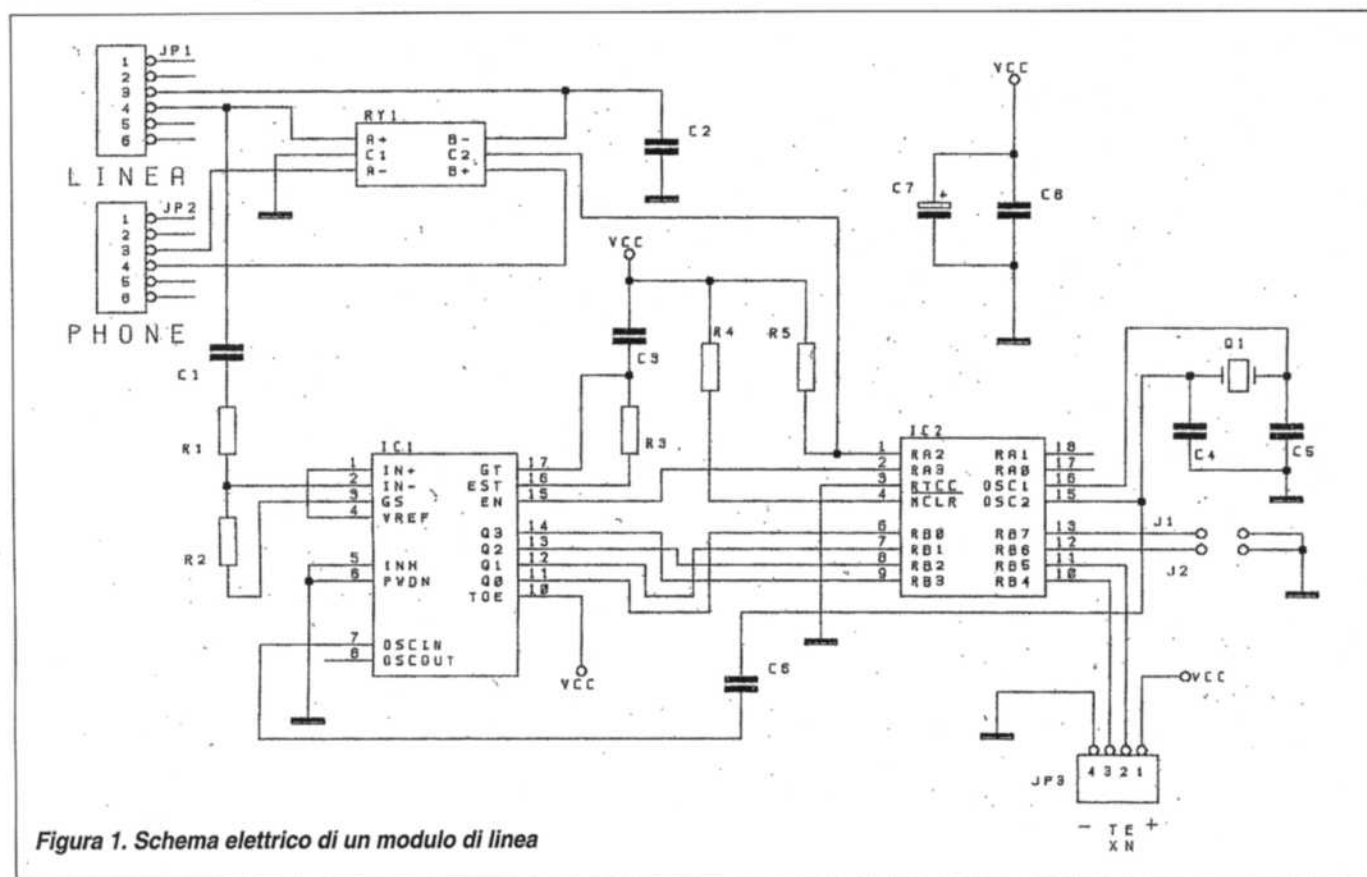


Figura 1. Schema elettrico di un modulo di linea

Per esempio, il numero 4 viene identificato con la frequenza relativa alla riga 2 sommata alla frequenza relativa alla colonna 1.

In questo modo con 4 frequenze di riga e 4 di colonna si ottengono 16 differenti composizioni.

Per riuscire a decodificare la combinazione delle due frequenze, sono disponibili sul mercato degli integrati che prendono in ingresso il segnale analogico e restituiscono in uscita il corrispondente valore digitale. In figura 2 vediamo l'interno dell'integrato che abbiamo impiegato per il nostro circuito.

Il segnale analogico entra in un amplificatore operazionale che lo amplifica per portarlo ad un livello adatto di comprensione.

Il blocco successivo è un filtro che elimina le frequenze superiori alla massima ammessa e le inferiori alla minima (cioè un filtro passa banda). Il segnale così filtrato, entra in altri due filtri passa banda più selettivi, che lo dividono nelle due frequenze di riga e di colonna.

A questo punto, le due frequenze passano attraverso un blocco di rilevamento digitale e vengono così riconosciute.

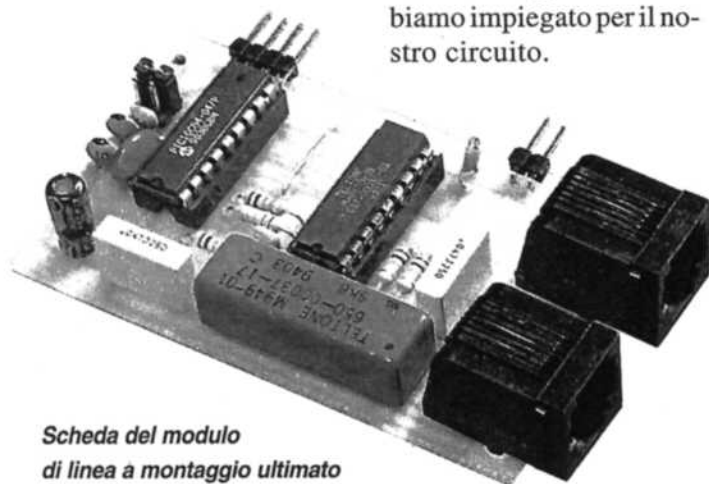
Una logica finale ne permette allora la presentazione verso il mondo esterno con bus three-state. I tempi di lavoro dell'8870 devono

essere molto precisi per la corretta individuazione delle frequenze da selezionare: per questo motivo la logica è sincronizzata con un oscillatore quarzato.

In figura 3 si vedono le varie fasi con i rispettivi tempi per la successione temporale degli eventi da riconoscere.

La sezione che si occupa di stabilire se la linea è aperta o chiusa oppure se il numero viene formato in modalità decadica è tutta centrata invece sul relè di linea M949-01, un componente della Teltonne progettato appositamente per questo scopo.

In figura 4 è visibile l'interno del relè: fondamentalmente è composto da due bobine a bassa resistenza



Scheda del modulo di linea a montaggio ultimato

che, in presenza di una minima corrente elettrica (4mA), creano un campo magnetico sufficiente a far attivare un piccolo contatto reed collocato tra di loro.

Quindi, quando la linea è aperta non c'è passaggio di corrente e conseguentemente il contatto è aperto, mentre quando la linea è chiusa il piccolo passaggio di corrente necessario a far funzionare il telefono crea il campo magnetico sufficiente a chiudere il contatto reed.

In questo modo, non solo individuamo lo stato della linea, ma anche se vengono formati degli impulsi sulla stessa e con quale durata. Naturalmente il calcolo delle tempistiche viene svolto dal microcontrollore IC2: si deve riuscire a capire se l'impulso letto è solo di apertura o chiusura linea oppure di formazione del numero e poi, in quest'ultimo caso, se è terminato il

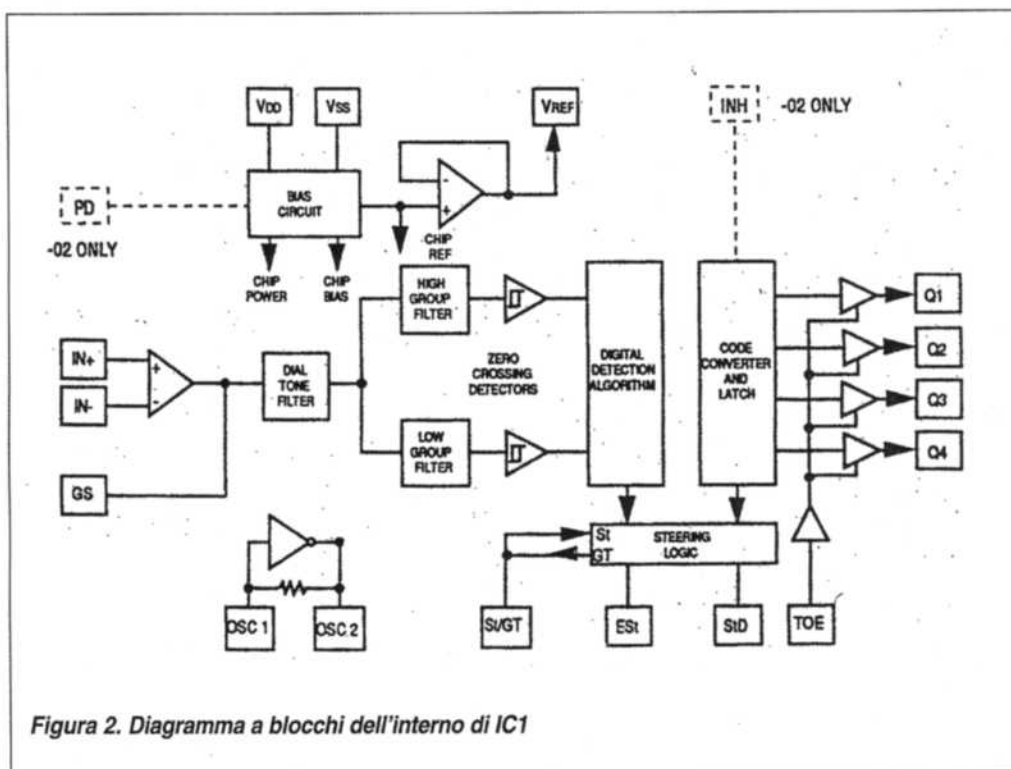


Figura 2. Diagramma a blocchi dell'interno di IC1

numero oppure no. Di tutto questo se ne occupa IC2, ovvero un PIC16C84 programmato per questa ed altre funzioni. IC2, infatti, è il cuore della sezione di logica e di controllo: le informazioni

che gli giungono sia dal relè di linea che da IC1 vengono elaborate e poi spedite su linea seriale. Per quest'ultima funzione però, la cosa non è identica al circuito precedentemente visto: poiché qui

possono essere presenti fino a 4 linee contemporaneamente, non è ammissibile che ognuna invii sulla porta seriale il segnale in qualsiasi momento.

Per questo motivo, la scheda ha un indirizzo selezionabile tra 0 e 3 in binario attraverso i due jumper J1 e J2 e, prima di trasmettere qualsiasi segnale, attende il "segnale di via libera" sul pin 11 di IC2. Il mese prossimo vedremo come avviene il dialogo tra scheda madre e schede di acquisizione.

A ognuno il proprio compito

Per capire meglio quali siano i compiti che deve svolgere il microcontrollore, valutiamo la figura 5 dove è possibile vedere il diagramma a blocchi relativo

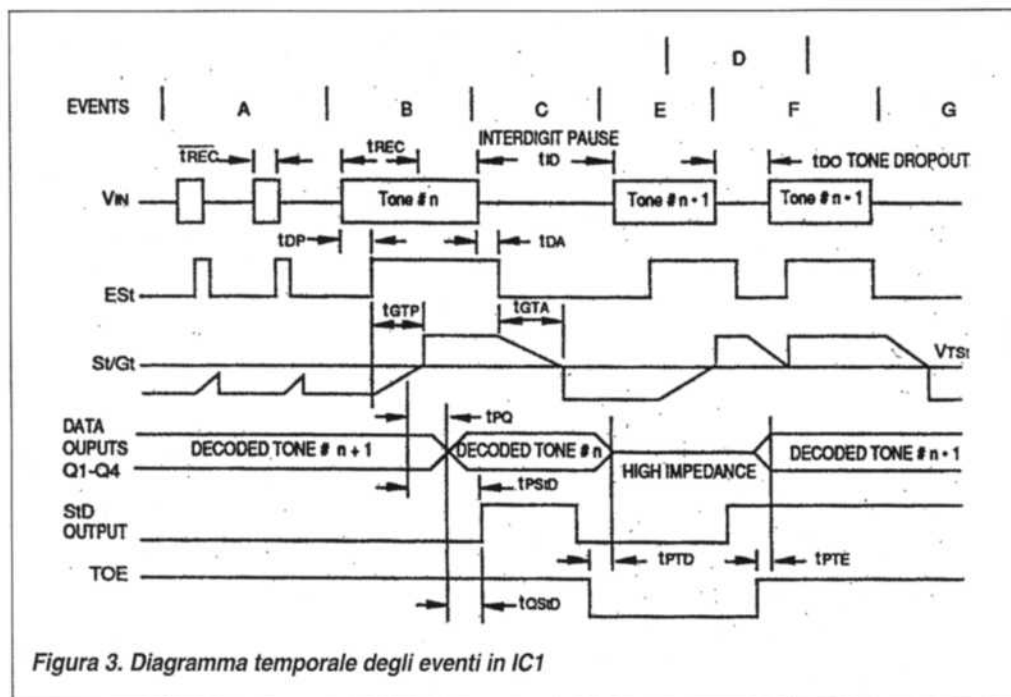


Figura 3. Diagramma temporale degli eventi in IC1

al firmware inserito in IC2 (ovviamente ogni scheda di acquisizione avrà lo stesso software). Normalmente il controller resta in attesa della chiusura della linea.

Quando ciò accade, la scheda invia, attendendo il suo turno, il carattere 0x1K sulla linea seriale dove K è il numero della scheda di acquisizione che può variare tra 1 e 4.

Successivamente il controller resta in attesa di uno dei tre seguenti eventi: numero in decadico, numero in DTMF, fine telefonata. Nel primo e nel secondo caso, il byte inviato sulla linea seriale sarà così composto: $0x(2 + K)0 + \text{cifra formata}$, con K come prima. Ad esempio, se la linea è la 3 e la cifra formata è il 7, il byte sarà $0x(2+3)0 + 7 =$

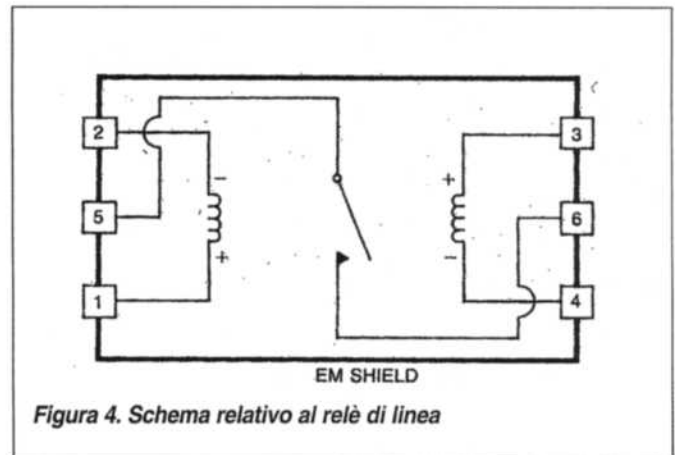


Figura 4. Schema relativo al relè di linea

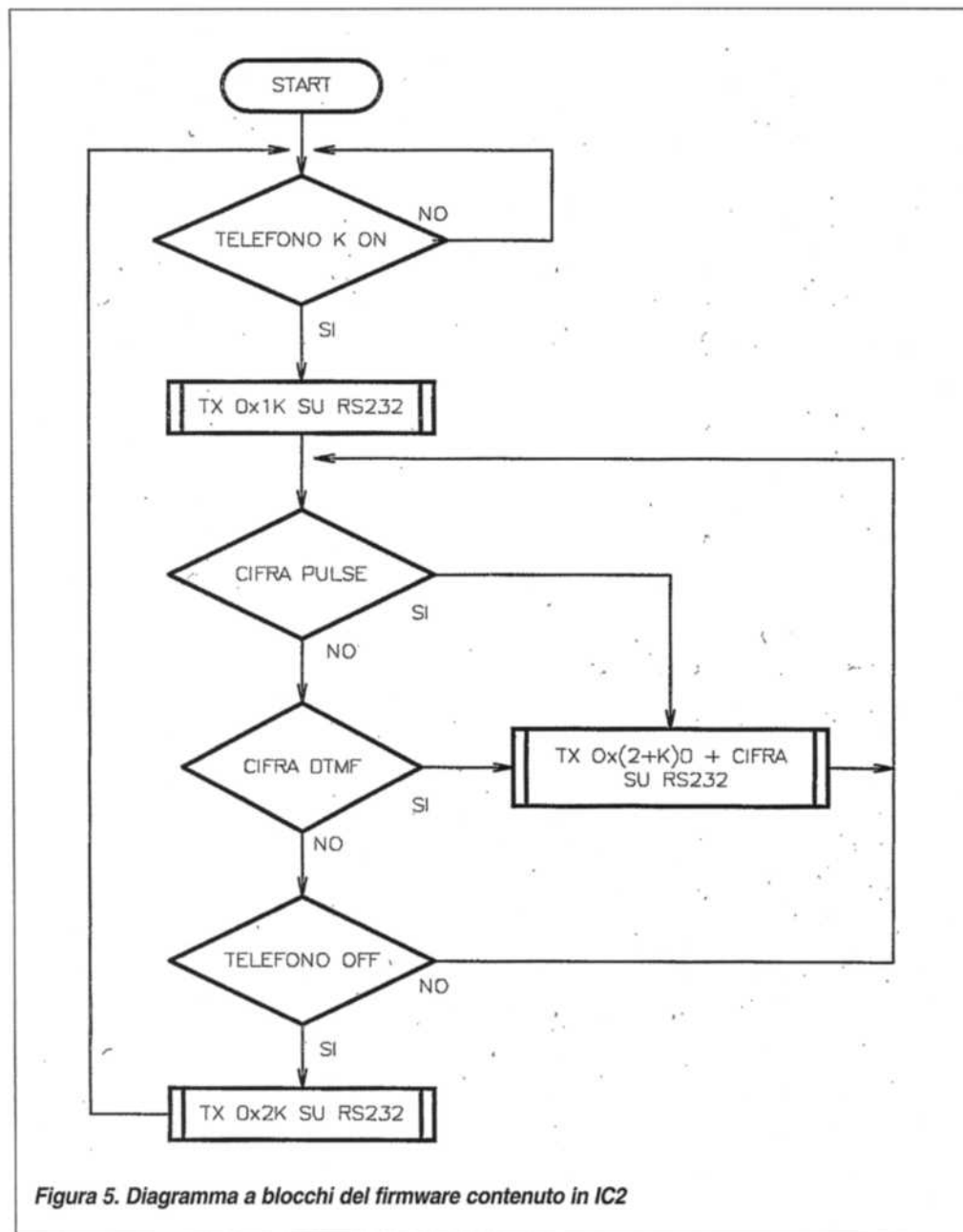


Figura 5. Diagramma a blocchi del firmware contenuto in IC2

0x57. Se invece la linea è la 1 e la cifra è sempre 7, avremo $0x(2+1)0 + 7 = 0x37$. Infine, il terzo evento previsto è la fine della telefonata che viene codificata con $0x2K$ dove K è sempre il numero della linea.

In questo modo, dal lato computer possiamo sempre sapere quale sia l'evento avvenuto e su quale linea è avvenuto.

Sarà poi compito del software applicativo sfruttare al meglio queste informazioni.

Il montaggio

La realizzazione delle schede (ricordiamo che ne occorre una per ciascuna linea telefonica) è piuttosto semplice, ricorrendo anche al disegno proposto in figura 6 per il circuito stampato.

Come al solito abbiamo preferito inserire un ponticello in filo di rame piuttosto che dover preparare uno stampato doppia-faccia. Prima di tutti gli altri componenti dovremo quindi inserire tale ponticello.

Passeremo poi agli zoccoli per i circuiti integrati ed alle resistenze.

Successivamente inseriremo tutti gli altri componenti di profilo sempre più alto.

Per connettere la scheda alla piastra madre, che vedremo il mese prossimo, consigliamo l'impiego di connettori a 90° per circuito stampato.

Per quanto riguarda le due prese telefoniche, dovranno essere ben saldate alla basetta, eventualmente aiutandosi con della colla a caldo, poiché saranno il punto di contatto con le linee telefoniche.

L'alimentazione per il circuito viene prelevata direttamente dal connettore verso la scheda madre: sono presenti quattro fili.

Due sono la massa ed il positivo di alimentazione (5 volt), uno è il segnale di abilitazione alla trasmissione seriale e l'ultimo è il segnale seriale vero e proprio. Ci sono poi due connettori che non sono collegati ad alcun filo, ma che servono

per rendere più stabile l'appoggio della scheda di acquisizione sulla scheda madre. Per collaudare la scheda ci sono due possibilità: la prima consiste nell'attendere il prossimo mese e realizzare la scheda madre, la seconda nel testare le varie funzioni servendosi di idonea attrezzatura.

Per iniziare, si deve collegare l'alimentazione alla scheda (rigorosamente 5 volt) e connettere la linea telefonica in una delle due prese.

All'altra presa collegheremo un classico telefono. Metteremo poi l'ingresso di enable al positivo di alimentazione (per simulare sempre la massima priorità di trasmissione) e poi dovremo interfacciare la linea seriale a 5 volt in uscita dal microcontrollore con la seriale RS232 del computer facendo però attenzione al fatto che sia optoisolata: in caso contrario non avremo il funzionamento richiesto, poiché l'interazione tra massa del computer e linea telefonica

ELENCO COMPONENTI

Semiconduttori

IC1: M8870
IC2: PIC16C84 programmato (0338/4890048)

Resistori

R1, R2: 100 kΩ
R3: 330 kΩ
R4, R5: 10 kΩ

Condensatori

C1, C2: 47 nF 160 V
C3, C8: 100 nF
C4, C5: 12 pF
C6: 33 pF
C7: 47 μF

Varie

RY1: Relè di linea M949-01
Q1: Oscillatore 3,579545 MHz

Prima di avviare il tutto si proceda alla selezione dell'indirizzo della scheda di acquisizione per mezzo dei jumper J1 e J2.

Entrambi connessi a massa selezionano l'indirizzo 0x00=0, mentre entrambi non connessi selezionano l'indirizzo 0x11=3.

Le posizioni intermedie identificano 0x01=1 e 0x02=2.

Fatto ciò, si avvia sul computer il software di dialogo con la porta seriale e si provi ad alzare la cornetta telefonica: sul monitor dovrà apparire il dato 0x1K.

Si provi poi a riabbassare la cornetta: sul monitor apparirà il carattere 0x2K.

Infine, formando dei numeri, questi dovranno essere visualizzati nella forma: 0x(2+K)0 + numero formato.

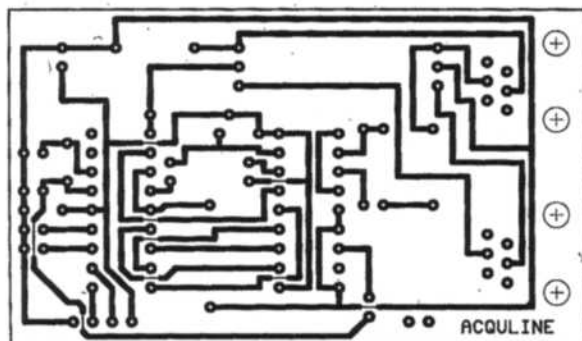
Il prossimo mese analizzeremo la scheda madre e vedremo il software applicativo proposto.

continua

Simone Argentini

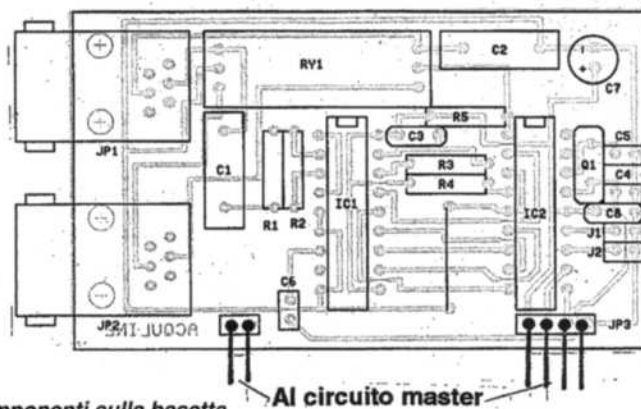
provocherebbe delle anomalie di funzionamento sulla normale linea telefonica.

Superato il problema dell'interfaccia optoisolata (ne abbiamo proposte già alcune) si dovrà dotare il computer di un software che visualizzi i dati che giungono sulla porta seriale.



Linea

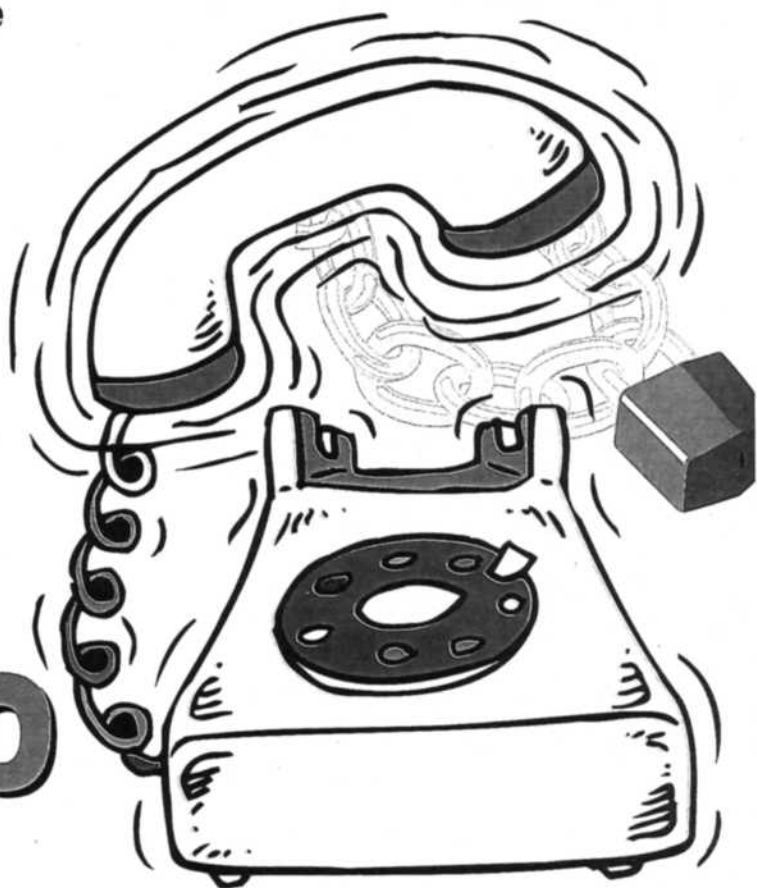
Telefono



Al circuito master

Figure 6 e 7. Circuito stampato scala 1:1 e disposizione dei componenti sulla basetta

Un centralino telefonico che registrerà tutte le chiamate uscenti con gestione di quattro linee telefoniche



Metti il telefono sotto controllo

Seconda parte

Continuiamo nella trattazione del sistema di controllo per telefonate uscenti iniziata lo scorso mese. Per coloro che non hanno avuto la possibilità di visionare la prima parte, ricordiamo brevemente quale sia il compito principale di questa apparecchiatura: supponiamo di avere in casa, in ufficio, in negozio, nella casa al mare una o più linee telefoniche

standard, cioè quelle che fino ad oggi la Telecom ha portato per mezzo di due fili intrecciati.

Fino a non molto tempo fa, tramite questi due fili era possibile collegare un solo telefono chiamante, un eventuale telefono solo ricevente ed eventualmente un apparecchio per l'ascolto della filodiffusione. Con l'arrivo massiccio dei computer, della tecnologia elettronica sempre più a basso costo e con l'aumento delle

esigenze di comunicazione, è adesso possibile avere in casa, collegati ai soliti due fili, un certo numero di apparati telefonici con diversi numeri, oltre al vantaggio di poter collegarsi ai circuiti telematici con modem che potenzialmente potrebbero arrivare a 128 kbps.

Naturalmente stiamo parlando di linee digitali o ISDN, cioè le nuove linee con flusso di dati a 64 kbps per canale.

di Simone Argentini

Inevitabilmente, con l'aumento delle linee a disposizione, aumenta anche il numero di telefonate effettuate nell'arco della giornata e, spesso, arrivano delle bollette che ci stupiscono largamente. In alcuni casi poi, si ha la necessità di distinguere da quale telefono siano uscite le chiamate più costose, per decidere se continuare a permettere tali chiamate oppure no.

Il sistema che proponiamo, riesce in gran parte a soddisfare tutte queste esigenze.

Innanzitutto è un sistema "componibile", ovvero composto da una piastra madre dove possono alloggiare fino a quattro schede (presentate lo scorso mese) ognuna delle quali deve essere connessa ad una linea telefonica (oppure ad un telefono ove ve ne siano più di uno sulla medesima linea).

Le postazioni che si possono controllare sono al massimo quattro.

Le informazioni ricevute dalla linea (o dal telefono) vengono passate ad un computer che le elabora e può, in seguito, visualizzarle e stamparle.

Il collegamento tra sistema di acquisizione e computer avviene tramite porta seriale RS232, quindi sarà possibile, per chi conosce un qualsiasi linguaggio di programmazione, realizzare un programma di gestione personale delle telefonate.

Noi ne abbiamo preparato uno scritto in Visual Basic 5 e quindi che gira solamente su computer con sistema operativo da Windows 95 in poi, ma è facilissimo scriverne uno in BASIC oppure in TurboPascal.

Il circuito elettrico della scheda madre

Dopo aver visto come sono realizzate e come funzionano le schede di acquisizione per ogni singola

linea telefonica, veniamo adesso alla descrizione della scheda madre, cioè a quella scheda dove devono essere inserite queste schede e da dove ci si deve collegare al computer.

In figura 8 abbiamo lo schema elettrico della scheda madre.

Si notano subito tre sezioni distinte: quella di controllo del bus, quella di interfaccia seriale e quella di alimentazione.

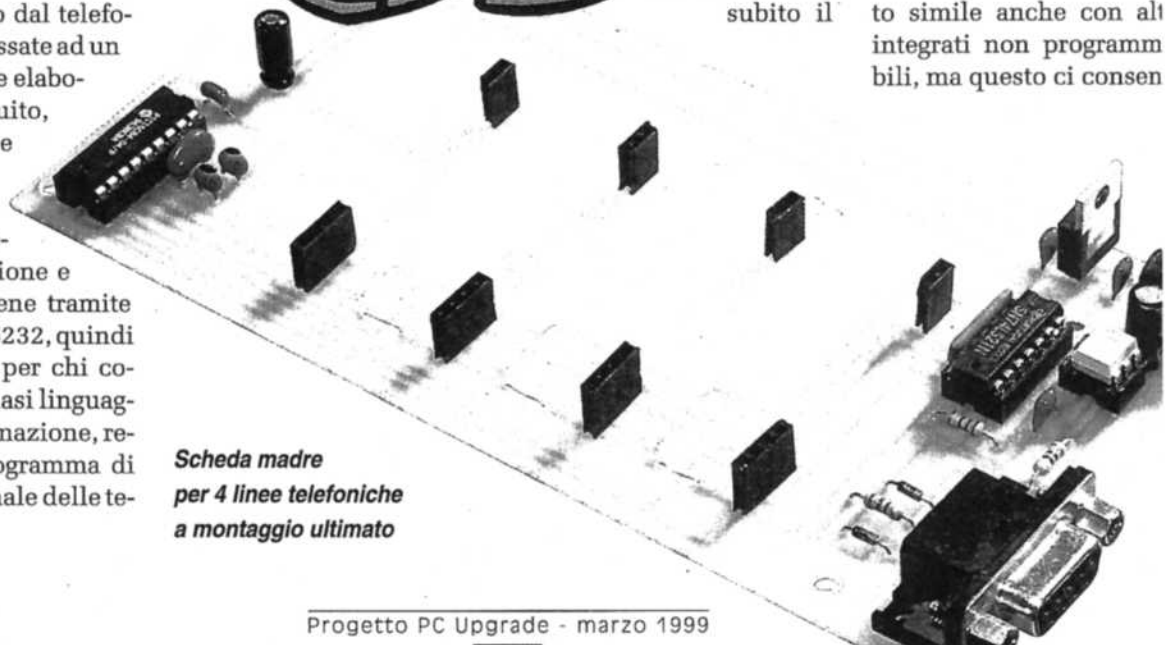
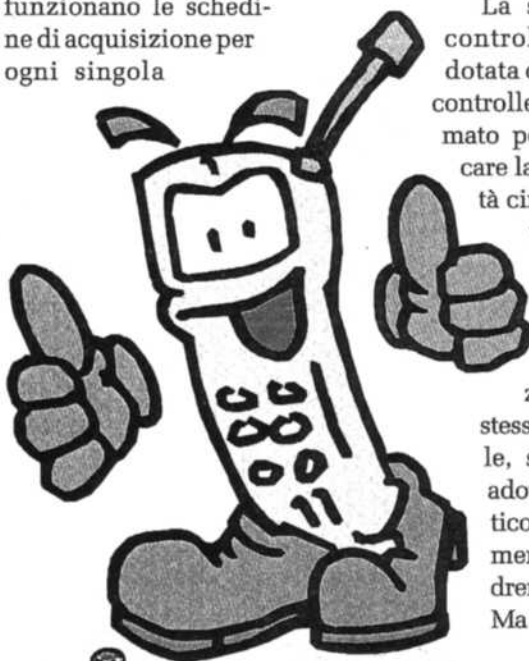
La sezione di controllo è stata dotata di un microcontroller programmato per semplificare la complessità circuitale: dovendo far dialogare tutte e 4 le schede di acquisizione sulla stessa porta seriale, si è dovuto adottare un particolare accorgimento che vedremo tra poco. Ma analizziamo subito il

firmware che è memorizzato in IC1 per avere un'idea di come lavori dando un'occhiata al diagramma a blocchi di figura 9: dopo la consueta e necessaria fase di inizializzazione dei registri e delle porte del controller si passa ad un'attesa di circa 20 millisecondi per consentire un reset a tutto il circuito. L'operazione successiva consiste nell'inviare un impulso di circa 100 microsecondi sul pin 4 della porta e poi di attendere circa millisecondi.

Questa operazione verrà ripetuta ciclicamente sul pin 5 della porta B, poi sul pin della stessa porta ed infine sul pin 7 sempre della stessa porta. Fatto ciò il ciclo ricomincia dal pin 4 della porta B e così via.

In pratica si sfrutta IC1 come generatore di impulsi di durata di circa 100 microsecondi con intervalli di circa 2 millisecondi ovvero come generatore di frequenza di circa 0,9 kHz con duty cycle del 5%.

Sarebbe stato possibile realizzare un funzionamento simile anche con altri integrati non programmabili, ma questo ci consen-



Scheda madre per 4 linee telefoniche a montaggio ultimato

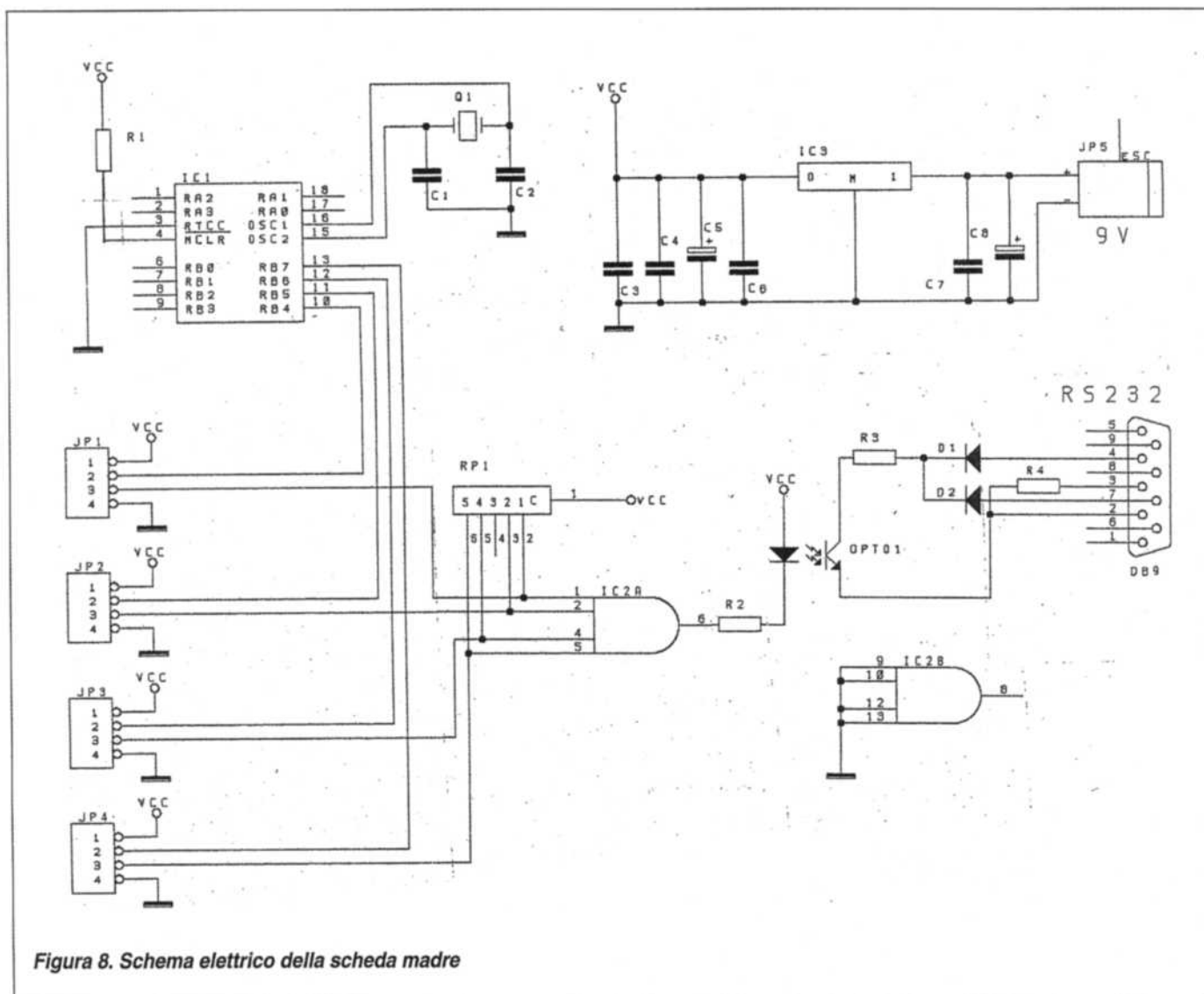


Figura 8. Schema elettrico della scheda madre

di avere sempre le stesse precise tempistiche e la possibilità, successivamente, di variare i parametri semplicemente riprogrammando il micro.

Per quanto riguarda l'interfaccia fisica con la porta seriale, abbiamo constatato che non era possibile realizzarla con i classici integrati tipo MAX232 poiché, venendo a contatto la massa del computer con quella delle linee telefoniche (anche se a mezzo di un condensatore),

si avevano delle interferenze sulle linee stesse, pregiudicandone il normale funzionamento.

Si è deciso così di optare per isolare la scheda madre dal computer (ottenendo così anche un risparmio economico perché un optoisolatore costa sulle 1.550 lire contro le 6/7.000 del MAX232!).

Per pilotare il led del fotoaccoppiatore, avevamo necessità di un buffer e così abbiamo scelto una porta AND a 4 ingressi che ci

consente di far arrivare sulla seriale uno qualsiasi dei segnali che gli giungono su uno degli ingressi.

Il motivo di ciò è molto semplice e va individuato nel funzionamento stesso di una linea seriale: normalmente, ovvero in assenza di segnale, la linea seriale resta ad alto livello (basso livello logico) ovvero il corrispondente segnale TTL o CMOS di pilotaggio deve essere a 5 volt.

Impiegando una porta AND, otteniamo sull'uscita un livello che sta a 5 volt solo quando tutti gli ingressi sono a 5 volt (nessuna trasmissione in corso) mentre passa a 0 volt ogni volta che un qualsiasi ingresso scende a tale tensione.

Sulla linea di trasmissione quindi, non si saprà mai chi è che ha trasmesso il segnale, ma l'importante è che questo giunga univocamente sulla linea, ovvero

non si deve verificare che due o più segnali si sovrappongano, pena la perdita di consistenza delle informazioni inviate.

L'ultima sezione da vedere è quella di alimentazione: un classico circuito regolatore a 5 volt che si occupa di fornire l'alimentazione anche alle singole schede di acquisizione e che pertanto dovrà essere in grado di supportare un carico di circa 100-200 mA senza sforzo.

Il dialogo tra scheda madre e schede di acquisizione

Vediamo adesso come dialogano la scheda madre con le quattro possibili schede di acquisizione.

Per prima cosa abbiamo già detto che sulla linea seriale di trasmissione verso il computer soltanto una scheda alla volta può chiaramente inviare dati.

Si deve poi tener presente che il protocollo tra scheda madre e computer prevede una velocità di 9.600 baud a 8N1 e che quindi, per l'invio di un byte, sono necessari $(1/9.600) * 10 = 1/960 = 1,04$ millisecondi.

Per capire come è stato fatto il conto, si devono valutare questi dati: innanzitutto per inviare un byte si devono inviare 10 bit e non 8 poiché si devono comunque conteggiare anche il bit di start ed il bit di stop.

Poi si deve calcolare il tempo necessario all'invio di un bit considerando la velocità di 9.600 bit al secondo che è di $1/9.600 = 104$ microsecondi.

A questo punto il conto è fatto: sono necessari circa 1,04 microsecondi per trasferire 10 bit ovvero un byte di dato.

Allora, per consentire a tutte le schede di inviare dati senza interferire l'una con l'altra, si è deciso di adottare la tecnica del polling: ogni scheda viene abilitata a trasferire dati circa ogni 8,4 millisecondi. Questo implica che le informazioni da trasferire non possono essere acquisite in tempi più frequenti di circa 10 millisecondi.

Ma per la nostra applicazione ciò è sufficiente poiché il tempo tra l'alzare la cornetta e comporre la prima cifra non sarà mai inferiore a 10 ms, come pure il tempo tra cifra e cifra di un numero.

Lo standard telefonico italiano prevede infatti che, nella modalità decadica, tra cifra e cifra passino almeno 200 ms, mentre nella modalità multifrequenza siamo intorno agli 80 millisecondi.

In teoria quindi, si potrebbero prevedere anche letture fino a 20 linee in contemporanea senza perdita di informazione e senza interferenza tra di loro.

In figura 10, si possono vedere i segnali di sincronismo generati da IC1 per le quattro schede di acquisizione.

Figura 9. Diagramma a blocchi del firmware inserito in IC1

