

L'interfaccia che oggi vi presentiamo, pur risultando molto meno complessa rispetto alle altre tanto che la relativa scheda ha dimensioni dimezzate, non è certo da meno come prestazioni in quanto non solo ci permette di pilotare una stampante, ma può essere utilizzata anche per infinite altre applicazioni soprattutto nel campo dei circuiti di controllo di macchine, sintetizzatori musicali, ecc.

Precisiamo subito che a tale interfaccia si possono collegare solo stampanti con ingresso «parallelo», cioè in pratica la **maggioranza** delle stampanti termiche o ad impatto attualmente di-

#### **1 ingresso e 1 uscita.**

In pratica, a differenza di quanto avveniva in precedenza, l'interfaccia memorizza il dato ad 8 bit che il microcomputer gli fornisce però anziché ripresentare in uscita contemporaneamente tutti gli 8 bit li invia in uscita uno per volta su un solo terminale.

Per esempio ammesso che il microcomputer fornisca all'interfaccia il codice 01110110, l'interfaccia ripresenterà sulla sua uscita i bit che compongono questo codice nella sequenza:  
prima condizione=0  
seconda condizione=1

## **INTERFACCIA** per

**Questa scheda oltre a risultare indispensabile per abilitare la nostra stampante a scrivere sulla carta i dati trasmessi dal microcomputer, può essere sfruttata anche per innumerevoli altre applicazioni, come per esempio il controllo di macchine utensili, in cui si abbia necessità di prelevare o fornire dei dati al microcomputer tramite un apposito circuito digitale esterno.**

sponibili sul mercato: ciò non toglie comunque che di queste stampanti ne esistano anche alcuni tipi provviste di solo ingresso «seriale», cioè con un ingresso totalmente diverso da quello «parallelo».

Tali stampanti, come vedremo meglio in seguito, non possono essere collegate a questa interfaccia, bensì ad un'apposita interfaccia con uscita «seriale» che presenteremo più avanti come scheda supplementare per consentire a chi disponesse eventualmente di questo tipo di macchina di poterla pilotare con il nostro microcomputer.

La differenza principale tra un'interfaccia con uscita di tipo «parallelo» ed una con uscita di tipo «seriale» può essere così riassunta:

#### **Interfaccia «parallelo»**

L'interfaccia dispone di **8 ingressi e 8 uscite** per i dati.

Quando il microcomputer vuole trasmettere un dato verso l'esterno, per esempio verso la stampante, lo trasmette all'interfaccia la quale lo memorizza nella propria memoria interna e contemporaneamente sulle 8 uscite presenta tutti gli 8 bit di questo dato, per esempio 01110110.

#### **Interfaccia «seriale»**

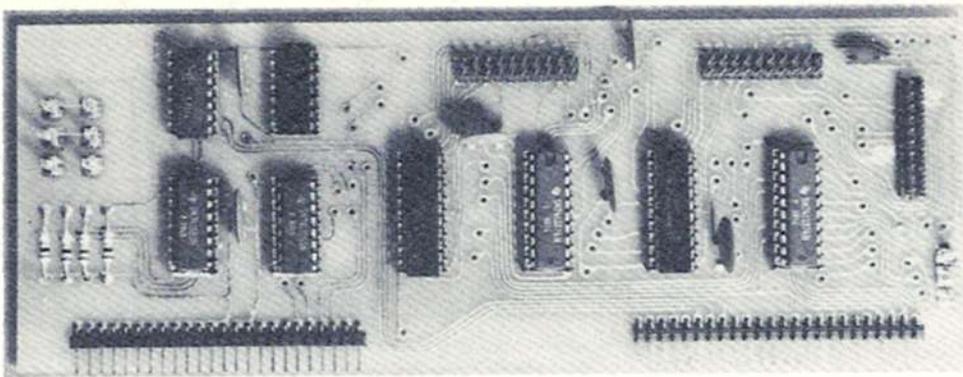
L'interfaccia seriale dispone invece per i dati di

terza condizione=1  
quarta condizione=0  
quinta condizione=1  
sesta condizione=1  
settima condizione=1  
ottava condizione=0

L'interfaccia aggiungerà inoltre, all'inizio di questa sequenza, un bit di «start» ed alla fine un bit di «stop» più uno di «parità» per far capire alla stampante o a qualsiasi altro circuito esterno che il dato è completo.

Una scheda con queste caratteristiche nel nostro microcomputer esiste già e precisamente trattasi della scheda LX385 impiegata per registrare i dati su nastro magnetico: su nastro infatti non è possibile registrare contemporaneamente gli 8 bit, bensì è necessario presentarli uno per volta in modo da poterli memorizzare uno dopo l'altro di seguito.

Vi ricordiamo comunque che la scheda «seriale» impiegata per il registratore a cassetta serve solo ed unicamente per svolgere questa funzione, cioè non è idonea per pilotare una stampante anche se di tipo seriale, infatti occorre tenere presente che tale interfaccia per distinguere la condizione logica 0 dalla condizione logica 1 fornisce in uscita due diverse frequenze rispettivamente di 1.200 Hz e di 2.400 Hz. Una stampante invece non vuole delle frequenze bensì richiede una tensione posi-



# STAMPANTE

tiva (normalmente di 5 volt) per la condizione logica 1 e una tensione nulla per la condizione logica 0.

In ogni caso, come vedremo sui numeri venturi, anche sull'interfaccia seriale è presente una UART (proprio come sull'interfaccia cassette) pur risultando questa scheda molto meno complessa dell'interfaccia cassette.

## Per le telescriventi

Finora abbiamo parlato di stampanti ma non di telescriventi ed è ovvio che chi dispone appunto di una telescrivente si chieda quale delle due interfacce può fare al caso suo.

Diremo subito che per questa applicazione è assolutamente necessaria l'interfaccia «seriale»: essendo però questa interfaccia meno indispensabile di quella di tipo «parallelo», abbiamo pensato di metterla per un momento in disparte dando la precedenza a ciò che ci sembrava più importante.

Ci riserviamo comunque di presentare molto presto anche questa interfaccia in modo tale che chiunque ne abbia necessità possa inserirla sul proprio BUS ed utilizzarla per le proprie specifiche applicazioni.

## SCHEMA ELETTRICO

Lo schema elettrico dell'interfaccia «parallelo», come vedesi in fig. 1, è molto semplice in quanto sono presenti solo 8 integrati.

Per quanto riguarda la descrizione di tale schema la prima cosa che vogliamo farvi notare è che questa scheda, a differenza dell'interfaccia video

che il microcomputer considera come un'espansione di memoria, è una «periferica» nel vero senso della parola proprio come l'interfaccia cassette, cioè tutte le operazioni di ingresso e uscita dati possono avvenire solo tramite un'istruzione di IN (cioè Input) o OUT (cioè Output) seguita dal codice che individua l'interfaccia stessa (ovviamente parliamo di istruzioni in linguaggio macchina).

In pratica lo scambio di dati fra la CPU e le **unità periferiche** come lo è appunto questa interfaccia avviene in modo molto simile a ciò che accade normalmente in un'azienda in cui vi sia un direttore ed un certo numero di segretarie che risiedono ciascuna in stanze diverse non comunicanti fra di loro però provviste di citofono collegato alla centralina del direttore.

Ciascuna segretaria (cioè ciascuna periferica) sarà contraddistinta da un proprio numero di individuazione (codice di indirizzo) e il direttore (cioè la CPU) avrà a disposizione sulla propria scrivania una tastiera con tutti i numeri con cui poter chiamare **una sola** delle sue segretarie senza disturbare le altre.

Se il direttore deve dettare una lettera (cioè fornire in uscita dei dati) alla segretaria n. 3, pigerà sulla propria scrivania il pulsante 3 ed automaticamente la segretaria 3, sentendosi chiamata al citofono, gli risponderà.

A questo punto il direttore le dirà: «Scrivi questa lettera» e non appena inizierà a dettare, la segretaria inizierà a scrivere a macchina; viceversa se il direttore le dirà: «leggimi le lettere che sono arrivate questa mattina», sarà la segretaria a dettare al direttore le informazioni in suo possesso.

Allo stesso modo si comporta la CPU del nostro microcomputer con le varie unità periferiche infatti quando vuole scrivere per esempio con la stam-



pante, deve prima mettersi in contatto con l'interfaccia «parallelo» fornendo il relativo codice di indirizzo sul BUS, poi fornire il dato all'interfaccia e dirle: «fa uscire questo dato verso la stampante».

L'interfaccia da parte sua memorizzerà questo dato all'interno di una propria unità di memoria e dalle uscite di questa lo trasferirà quindi alla stampante che provvederà a scriverlo sulla carta.

Viceversa quando la CPU vuole ricevere in ingresso un dato per esempio da un circuito esterno che noi abbiamo collegato all'entrata 2 dell'interfaccia «parallelo» come prima cosa deve sempre mettersi in contatto con tale interfaccia fornendo il relativo codice di indirizzo poi deve dire a questa interfaccia, sempre tramite un codice binario:

#### COMPONENTI

C5 = 47.000 pF a disco  
 C6 = 47.000 pF a disco  
 C7 = 47.000 pF a disco  
 IC1 = integrato tipo SN74LS155  
 IC2 = integrato tipo SN74LS04  
 IC3 = integrato tipo SN74LS02  
 IC4 = integrato tipo SN74LS85  
 IC5 = integrato tipo SN74LS374  
 IC6 = integrato tipo SN74LS244  
 IC7 = integrato tipo SN74LS374  
 IC8 = integrato tipo SN74LS244

R1 = 10.000 ohm 1/4 watt  
 R2 = 10.000 ohm 1/4 watt  
 R3 = 10.000 ohm 1/4 watt  
 R4 = 10.000 ohm 1/4 watt  
 R5 = 10.000 ohm 1/4 watt  
 C1 = 47.000 pF a disco  
 C2 = 47.000 pF a disco  
 C3 = 47.000 pF a disco  
 C4 = 47.000 pF a disco

Fig. 1 Schema elettrico dell'interfaccia per stampante completa di entrate e uscite supplementari che potremo utilizzare per realizzare dei servocontrolli gestiti con programmi tramite il microcomputer.

Nota: se si utilizza una sola stampante i ponticelli P1-P2-P3 debbono essere cortocircuitati. Con taluni tipi di stampanti ad impatto, il terminale 17 (uscita stampante) deve essere collegato a massa.

«passami il dato presente sull'entrata 2».

A questo punto l'interfaccia abiliterà l'integrato IC8 il quale contiene al suo interno 8 interruttori elettronici (anche se il termine non è dei più appropriati) a chiudere questi interruttori ed a lasciar passare verso la CPU il dato applicato dall'esterno sull'entrata 2.

Quanto abbiamo appena accennato dovrebbe subito farvi comprendere la grande importanza di questa interfaccia «parallelo» nell'ambito del microcomputer infatti questa non solo serve per pilotare la stampante ma ci permette anche di sfruttare tutta la potenza di questa macchina per realizzare dei servocontrolli eccezionalmente precisi e affidabili.

Tanto per rendervi un'idea noi potremmo sfruttare le uscite di questa interfaccia per eccitare dei

relè in una sequenza ben determinata in modo da comandare dei motorini, contatori, ventilatori o altre apparecchiature elettroniche ad intervalli regolari stabiliti tramite un programma.

Servendoci di opportuni trasduttori potremmo tener sotto controllo la posizione di un «asse» in una macchina utensile e far compiere a questo «asse» degli spostamenti ben determinati.

Servendoci di altri trasduttori potremmo invece tener sotto controllo una temperatura, il numero di giri di un motore, la pressione in una pompa ecc. oppure modificare a piacimento il ciclo produttivo di una macchina semplicemente modificando il programma all'interno del microcomputer.

Supponiamo per esempio di utilizzare l'uscita 1 dell'interfaccia e di collegare ai terminali d'uscita 2-3-4 (corrispondenti a D0-D1-D2) tre relè che pilotano rispettivamente un motorino, una pompa e un ventilatore.

Se noi volessimo realizzare un ciclo di lavorazione in cui si attiva innanzitutto il motorino per 3 secondi, poi si ferma il motorino e si accendono il ventilatore e la pompa per 5 secondi, poi si ferma il ventilatore e resta accesa la pompa per altri 2 secondi, non dovremmo fare altro che scrivere in memoria un programma che manda fuori innanzitutto sull'uscita 1 (vedremo poi il relativo codice di accesso) il dato:

**10000000** (si mette in moto il motore)

A questo punto, sempre nel programma, dovremmo realizzare una «routine» di ritardo di 3 secondi, per esempio sfruttando la subroutine 8094 (vedi rivista 71 a pag. 74) già presente nella Eprom, poi mandare di nuovo fuori sull'uscita 1 il dato:

**01100000** (si accendono ventilatore e pompa)

Dopo di ciò occorrerebbe ancora una routine di ritardo, per esempio la subroutine 80E3 che determina un ritardo fisso di 5 secondi e trascorsi questi 5 secondi dovremmo mandare fuori sempre sull'uscita 1 il dato:

**01000000** (resta accesa la sola pompa)

Ancora ci servirebbe una routine di ritardo di 2 secondi che potremmo ottenere per esempio con la subroutine 808E e trascorsi questi 2 secondi dovremmo inviare sull'uscita 1 il dato:

**00000000** (tutto spento)

Ammesso che in un secondo tempo si rendesse poi necessario modificare questo ciclo, noi non dovremmo cambiare nessun circuito elettrico ma solo ed esclusivamente il programma in memoria.

Per quanto concerne le entrate dell'interfaccia queste possono essere sfruttate per acquisire dati dall'esterno, dati che ovviamente verranno utilizzati nel corso di un programma per ottenere degli scopi ben determinati.

Per esempio noi potremmo collegare su queste entrate un commutatore binario con il quale fornire al microcomputer le «quote» iniziali di un determinato spostamento, oppure le uscite binarie di un qualsiasi trasduttore di posizione, temperatura ecc. che fornisca al microcomputer le informazioni

che gli necessitano nel corso di un programma da noi impostato.

Come vedete, le applicazioni sono molteplici e dipenderà solo dalla fantasia dell'utilizzatore trovarne tante altre perfino insospettite.

Premesso questo possiamo ora dedicare la nostra attenzione al codice di «indirizzo» di questa scheda, cioè quella particolare combinazione binaria che ogniqualvolta viene presentata dalla CPU sui relativi terminali del BUS, fa capire alla periferica che si vogliono trasmettere dei dati in uscita o ricevere dei dati in ingresso.

A tale proposito precisiamo subito che il codice di indirizzo della scheda non è «unico», bensì per la presenza dei ponticelli P1-P2-P3 che troviamo sugli ingressi di destra dell'integrato IC4, vi sono ben 8 diverse combinazioni che possono essere assegnate a questa scheda.

Poiché come abbiamo visto sono proprio questi terminali che determinano il nostro indirizzo, giustamente il lettore si chiederà il motivo per cui sono stati previsti questi ponticelli sullo stampato, ponticelli che ovviamente ci permettono di modificare tale indirizzo a nostro piacimento.

La risposta è molto semplice infatti se si voleva realizzare un microcomputer che fosse il più versatile possibile, non si poteva certo obbligare il lettore a collegare una sola stampante oppure a disporre di una sola uscita «parallelo» perché possono esservi applicazioni in cui è richiesto l'uso di più di una stampante contemporaneamente ed altre applicazioni in cui sono richieste per esempio 10-12 diverse uscite di tipo parallelo.

Per non limitare il lettore nella sua creatività abbiamo quindi previsto di poter assegnare a questa scheda diversi codici di indirizzo in modo da poter aggiungere alla stampante ufficiale altre stampanti di servizio (per esempio potremmo impiegarne contemporaneamente una termica per le applicazioni di minor impegno e una ad impatto per le fatture, tabulati ecc.) oppure da poter comandare con un solo microcomputer 5-6 apparecchiature

elettroniche diverse tenendole tutte sotto controllo contemporaneamente.

Essendo 3 i ponticelli previsti ci saranno permesse un massimo di 8 combinazioni diverse, cioè potremo utilizzare un **massimo di 8 schede** per un totale di 16 entrate e 16 uscite «parallelo» (infatti ogni scheda dispone di 2 entrate e 2 uscite); il numero massimo di stampanti che si possono pilotare è invece limitato a 8.

Inutile aggiungere che sarebbe piuttosto noioso andarsi ora a determinare uno per uno gli indirizzi che individuano tutti questi ingressi e uscite pertanto abbiamo ritenuto opportuno condensare in tabella n. 1 questi codici onde facilitarne la ricerca e l'impiego nei programmi.

A proposito di questa tabella dobbiamo qui aprire una piccola parentesi **per chiarire che i codici di indirizzo** in essa riportati ci serviranno **solo** quando programmeremo con la **tastiera esadecimale** in quanto utilizzando la tastiera alfanumerica il procedimento per individuare uno qualsiasi di questi ingressi o uscite, come vedremo presentandovi il linguaggio BASIC, è totalmente diverso.

Programmando in esadecimale tali codici di indirizzo vanno sempre abbinati a un'istruzione IN (cioè ingresso) o OUT (cioè uscita): per esempio, ammesso di aver effettuato **tutti e tre i ponticelli** sulla scheda, se noi volessimo trasferire all'uscita 1 il numero contenuto nel registro A, dovremmo utilizzare l'istruzione **OUT(03),A** che in esadecimale si scrive:

**D3**

**03** (03 è l'indirizzo dell'uscita 1)

se invece volessimo compiere l'operazione inversa cioè caricare nel registro A il numero applicato dall'esterno sull'ingresso 2 della scheda «parallelo», l'istruzione che dovremmo utilizzare sarebbe **IN A,(02)** che in esadecimale si scrive:

**DB**

**02** (02 è l'indirizzo dell'entrata 2).

Viceversa se sulla scheda avessimo effettuato il solo ponticello P2, vedi riga 6 della tabella, l'istru-

Tabella n. 1

Ponticelli effettuati			Indirizzo entrata-uscita 1 e stampante	Indirizzo entrata-uscita 2
P1	P2	P3		
SI	SI	SI	03	02
SI	SI	NO	07	06
SI	NO	SI	0B	0A
SI	NO	NO	0F	0E
NO	SI	SI	13	12
NO	SI	NO	17	16
NO	NO	SI	1B	1A
NO	NO	NO	1F	1E

zione che dovremmo utilizzare nel primo caso sarebbe **OUT(17),A** che in esadecimale si scrive:

**D3**

**17** (17 è l'indirizzo dell'uscita 1)

mentre nel secondo caso sarebbe **IN A,(16)** che in esadecimale si scrive:

**DB**

**16** (16 è l'indirizzo dell'entrata 2).

Prima di concludere questa nostra descrizione per passare alla realizzazione pratica vogliamo farvi notare che sul connettore riservato alla stampante sono presenti, a parte i terminali di massa, due terminali aggiuntivi rispetto agli 8 terminali dei dati e precisamente abbiamo un terminale indicato con **STROBE** ed uno con **BUSY**.

Il primo di questi terminali ci servirà, quando vorremo far stampare un dato tramite la tastiera esadecimale, per comunicare alla stampante che il dato presente in uscita è **valido** infatti la stampante stessa prima di accettare in ingresso qualsiasi dato, richiede sempre che gli si mandi un impulso negativo sul terminale di **STROBE**, diversamente ignora i dati da noi forniti.

Questo impulso negativo lo si ottiene da programma con un'istruzione di **SET** seguita da un **RES** e da un nuovo **SET** relativo al bit 7 del dato in uscita.

Il **BUSY** è invece un segnale che viene inviato **dalla stampante al microcomputer** quando essa è impegnata a scrivere per dirgli di non mandare altri dati.

In pratica volendo comporre un programma per far stampare qualcosa alla stampante tramite la tastiera esadecimale, noi dovremo sempre mettere innanzitutto un'istruzione di **INPUT** dall'entrata 1 seguita da un test sul bit 0 per controllare che il **BUSY** non sia attivo (cioè in condizione logica 1) diversamente anche se noi inviassimo dati alla stampante, questa non potrebbe accettarli perché già impegnata a scrivere.

Non ci dilunghiamo comunque su tale argomento in quanto a fine articolo il lettore troverà un programma in esadecimale da cui potrà facilmente capire qual è la prassi seguita ogni volta per far stampare un carattere.

## REALIZZAZIONE PRATICA

Il circuito stampato necessario per questa realizzazione porta la sigla **LX389** e come tutti i circuiti stampati di questo microcomputer è un **doppia faccia a fori metallizzati**, vale a dire che tutti i collegamenti fra le piste superiori ed inferiori sono già stati effettuati per via galvanica in fase di incisione pertanto non si richiede da parte nostra di eseguire nessun ponticello con filo di rame.

Vi ricordiamo comunque di non tentare per nessun motivo di allargare questi fori diversamente asportereste il rame interrompendo così il collegamento elettrico fra le piste.

Una volta in possesso di tale circuito monteremo subito gli zoccoli per gli integrati poi tutte le resistenze e i 5 connettori, cercando di utilizzare uno stagnatore a punta fine e soprattutto facendo attenzione a sciogliere sempre il minimo di stagno indispensabile per non creare dei cortocircuiti fra i terminali tutti molto vicini fra di loro.

Per ultimi monteremo i condensatori a disco dopodiché potremo inserire sui relativi zoccoli i vari integrati rispettandone la tacca di riferimento ed a questo punto la nostra interfaccia sarà già pronta per essere inserita sul BUS: prima comunque dovremo ricordarci di effettuare i **tre ponticelli P1-P2-P3** in modo da assegnare alla scheda il codice di indirizzo 03) per l'ingresso e uscita 1 e il codice 02 per l'ingresso e uscita 2, come appunto si richiede per poter accedere alla stampante con il **BASIC**.

## PER PROVARE LA SCHEDA

Una volta terminato il montaggio è chiaro che tutti avranno la curiosità di collaudare la scheda per vedere finalmente la loro stampante scrivere un qualcosa di diverso dal semplice «self-test».

In previsione di questo vi abbiamo preparato un semplice programma che unisce per così dire l'utile al dilettevole infatti questo non solo vi permette di stampare qualcosa ma vi evita anche tanta fatica in quanto vi permette di stampare il «listing» esadecimale di un qualsiasi programma presente in memoria, compreso il listing del programma «monitor» residente nella Eprom che tanti lettori ci hanno richiesto.

Tale programma, nella versione in cui lo presentiamo, deve essere scritto in memoria a partire dalla riga **0070** ed occupa fino alla riga **00EA** pertanto la sua unica limitazione è quella di non poter «listare» un altro programma che occupi la sua stessa area di memoria.

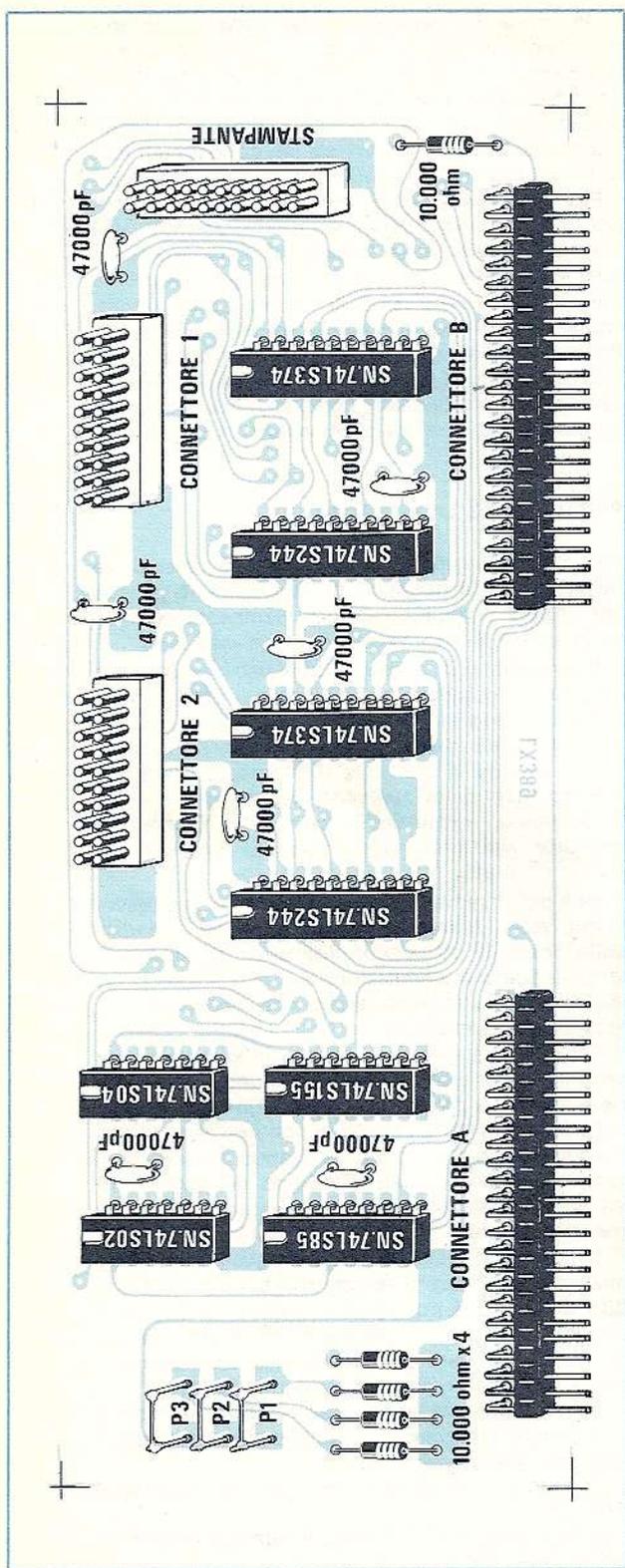
Per esempio se noi avessimo già in memoria un programma che inizia dalla locazione **0070**, come per esempio il programma «Magazzino» presentato sulla rivista n. 71, è ovvio che non potremmo farne il «listing» totale in quanto scrivendo in memoria il programma che ci serve appunto per fare il listing, automaticamente cancelleremo una parte del precedente.

Se invece il programma che vogliamo stampare inizia per esempio dalla riga **0100** oppure dalla riga **0200** è ovvio che non esiste il problema in quanto anche scrivendo in memoria il programma di «listing» non arriveremo mai alla riga **0100**, bensì ci fermeremo, come già detto, alla riga **00EA**.

La procedura da seguire per scrivere in memoria il nostro programma è quella solita che ormai vi abbiamo ripetuto tante volte:

1) pigiate sulla tastiera esadecimale il pulsante **RESET**.

2) Impostate sulla tastiera il numero **0070** e pigiate i due tasti **CONTROL-0**.



3) Sui display 0-1 vi appariranno due numeri casuali che rappresentano il vecchio contenuto della riga 0070.

4) Impostate sulla tastiera il codice CD quindi pigiate i due tasti CONTROL-0 per trasferire tale numero in memoria.

5) Sui display vi apparirà l'indirizzo successivo, cioè 0071 seguito ancora da due numeri casuali.

6) Impostate il numero 67 sulla tastiera e pigiate ancora CONTROL-0.

7) Continuate ad inserire i dati riportati nella tabella, cioè proseguite con 80-CD-EC-80-3E ecc. fino ad arrivare alla riga 00EA in cui dovrete memorizzare un C9 ed a questo punto ricontrollate tutto il vostro programma per vedere di non aver commesso errori.

Terminato di scrivere questo programma, per farlo eseguire dovrete caricare nel registro PC (Program Counter) il numero 0070 che corrisponde appunto alla prima riga del programma stesso.

Per ottenere ciò dovrete procedere come segue:

1) Pigiate i due tasti CONTROL-2 per accedere ai registri.

2) Pigiate tante volte di seguito i due tasti CONTROL-0 finché non vi verrà visualizzato il contenuto del registro PC sui display.

3) Impostate sulla tastiera il numero 0070 quindi pigiate i due tasti CONTROL-0 per trasferire tale indirizzo in memoria.

Eseguita anche questa operazione il vostro programma è pronto per essere fatto «girare», quindi potrete tranquillamente pigiare i due tasti CONTROL-4.

**Fig. 2** Schema pratico di montaggio dell'interfaccia parallela. Sul connettore 1 fanno capo i terminali d'uscita e d'entrata relativi agli integrati IC5-IC6 mentre sul connettore 2 i terminali d'uscita e d'entrata relativi agli integrati IC7-IC8.

La stampante dovrà essere collegata al connettore posto in verticale sulla destra.

Automaticamente sui display vi apparirà:



In pratica il computer vi chiederà l'indirizzo di partenza della vostra stampa (è ovvio che la stampante dovrà già essere collegata sull'apposito connettore e soprattutto dovrà essere accesa).

Ammesso per esempio di voler fare il listing della «subroutine» 80AA del «monitor» (che inizia alla riga 80AA e termina alla riga 80D5) dovrete a questo punto impostare sulla tastiera il numero 80AA, in modo da far apparire sui display:



Controllate che tutto sia esatto poi pigiate i due tasti **CONTROL-0** per memorizzare questo indirizzo di partenza.

Automaticamente il computer vi chiederà l'**indirizzo finale** della stampa facendo di nuovo comparire sui display la scritta:



Impostate sulla tastiera il numero 80D5 in modo da far apparire sui display:



Controllate che tutto sia esatto poi pigiate i due tasti **CONTROL-1**.

Immediatamente la stampante inizierà ad eseguire il listing di questa subroutine o di qualsiasi altro programma che voi vogliate stampare e quando arriverà alla fine si fermerà e sui display della tastiera tornerà a comparire la scritta:



Questa significa che il computer è pronto per eseguire un secondo «listing» e vi chiede il nuovo indirizzo di partenza.

Se avete un secondo listing da eseguire impostate l'indirizzo di partenza come avevate fatto in precedenza, pigiate i due tasti **CONTROL-0**, poi impostate l'indirizzo finale e pigiate i due tasti **CONTROL-1**.

Se invece non avete più nessun listing per fermare il computer vi basterà pigiare il pulsante di **RESET**.

**Nota importante:** per poter utilizzare il programma precedente è assolutamente necessario che sull'interfaccia parallela siano stati eseguiti tutti e **tre i ponticelli P1-P2-P3**, cioè che all'uscita «stampante» sia stato assegnato il codice di indirizzo 03.

Qualora per motivi personali alla stampante stessa avessimo assegnato un diverso codice, per esempio 0B o 1F (vedi tabella precedente), nel programma dovremmo modificare le seguenti righe:

00C3=nuovo codice (per esempio 0B o 1F)

00CD=nuovo codice

00D1=nuovo codice

00D5=nuovo codice

## PER SPOSTARE IL PROGRAMMA IN MEMORIA

Precisiamo subito che questo paragrafo è riservato ai più esperti quindi non ce ne vogliate se saremo un po' troppo «tecnici» come linguaggio.

Ammettiamo che a qualcuno interessi eseguire con questo programma di stampa il listing di un programma che inizia dalla riga 0070.

Vi abbiamo già detto che nella versione attuale un'operazione di questo genere è impossibile però si può ovviare all'inconveniente in modo molto semplice scrivendo il programma di stampa in

un'altra zona di memoria che risulti libera, per esempio partendo dalla riga 1070 invece che dalla 0070.

Così facendo è ovvio che la riga 0071 diventerà la 1071, la riga 0072 diventerà la 1072 e così di seguito.

Anche il Program Counter (registro PC) dovrà essere modificato e precisamente su questo registro, invece di caricare il numero 0070, dovremo caricare **1070**.

Le istruzioni rimarranno tutte invariate fatta eccezione per le istruzioni **CALL** riferite a subroutine che non risultino incluse nel programma «monitor», cioè **CALL 00B7**, **CALL 00C1** e **CALL 00D8**.

Queste istruzioni, ammesso sempre di scrivere il programma a partire dalla riga 1070 invece che dalla 0070, dovranno essere modificate come segue:

**CALL 00B7** diventerà **CALL 10B7**

**CALL 00C1** diventerà **CALL 10C1**

**CALL 00D8** diventerà **CALL 10D8**.

In pratica le righe di memoria interessate da questi cambiamenti saranno:

1096=10 invece di 00

109A=10 invece di 00

109F=10 invece di 00

10A2=10 invece di 00

10A6=10 invece di 00

10AB=10 invece di 00

10E1=10 invece di 00

10E8=10 invece di 00.

Così facendo potremo eseguire il listing anche del programma «magazzino» sebbene questo inizi dalla riga 0070.

## LA SUBROUTINE DI STAMPA

Sempre per i più esperti vogliamo infine spendere due parole riguardo la subroutine da noi realizzata per la trasmissione dati alla stampante, subroutine che inizia alla riga 00B7 e termina alla riga 00D7.

Questa subroutine può essere utilizzata anche in altri programmi diversi dal nostro ricordandosi ogni volta di caricare nel registro A il numero che si vuole stampare, cioè 01-02-03-...-0A-0B ecc. quindi chiamare la subroutine stessa con un'istruzione **CALL 00B7**.

Automaticamente il numero contenuto nel registro A verrà convertito in codice ASCII e trasmesso alla stampante, che momentaneamente lo conserverà nella propria memoria interna, senza stamparlo, mettendolo in coda agli altri caratteri già ricevuti in ingresso.

Quando vorremo stampare il tutto non dovremo fare altro che caricare nel registro A il numero 0D e senza convertire questo codice in ASCII, passarlo subito alla stampante chiamando la subroutine 00C1, cioè **CALL 00C1**.

Per lasciare uno spazio bianco fra un carattere ed il successivo nella stampa si deve invece cari-

care nel registro A il numero 20 e trasmetterlo quindi alla stampante ancora con la subroutine 00C1.

## CONCLUSIONE

Quello che vi abbiamo appena presentato è solo un esempio di come si possa «pilotare» la stampante con il nostro microcomputer per fargli stampare un qualche cosa (nella fattispecie il listing di un programma), tuttavia dovrebbe servirvi per comprendere che una volta in possesso della stampante stessa e della relativa interfaccia, non è sufficiente collegarle insieme e pigiare quindi un tasto per vedere il relativo carattere stampato sulla carta.

Ogni operazione che il nostro microcomputer esegue è infatti sempre un'operazione che gli è stata ordinata tramite un programma e questo vale per la stampante, come per il monitor o qualsiasi altra periferica collegata al microcomputer stesso.

Possiamo comunque anticiparvi che una volta caricato nel microcomputer il linguaggio BASIC (vedi apposito articolo su questa stessa rivista) tutto diventa più facile: per esempio il listing di un programma si può ottenere semplicemente fornendo al computer l'istruzione SLISTA; la stampa

di una frase o di un numero si può ottenere semplicemente con l'istruzione SSCRIVI, cioè basta una sola parola per far capire al computer ciò che deve fare.

Non spaventatevi quindi vedendo un programma apparentemente così complesso per ottenere una funzione che ritenevate elementare ed automatica: tale programma infatti serve solo a chi non utilizza la tastiera alfanumerica e il BASIC perché nel BASIC lo stesso programma è condensato in una sola frase.

## COSTO DELLA REALIZZAZIONE

Il solo circuito stampato LX389 in fibra di vetro a doppia faccia con fori metallizzati L. 12.500

Tutto il materiale occorrente, cioè circuito stampato, resistenze, condensatori, integrati e relativi zoccoli, connettori L. 49.000

Una piattina lunga 1 metro con due connettori agli estremi per collegarsi alla stampante termica presentata sul n. 72 L. 10.000

I prezzi sopra riportati non includono le spese postali.

### Programma: LISTING su STAMPANTE

Riga	Codice	Mnemonico	Descrizione	Commento
0070 0071 0072	CD 67 80	CALL 8067	Esegui la subroutine 8067 (vedi riv. 71)	Spegne tutti i display facendo comparire una L. nel primo display a sinistra.
0073 0074 0075	CD EC 80	CALL 80EC	Esegui la subroutine 80EC	
0076 0077	3E 47	LD A,47	Carica in A il numero 47 corrispondente sul display a una L.	
0078 0079	D3 F7	OUT(F7),A	Trasmetti il contenuto di A al display 7	Aspetta che noi pigiamo un tasto qualsiasi; controlla se per caso abbiamo pigiato CONTROL-0 oppure CONTROL-1; se non è stato pigiato né CONTROL-0 né CONTROL-1 ma un tasto qualsiasi fa apparire il relativo numero sui display spostando di una posizione verso sinistra le cifre già presenti.
007A 007B 007C	CD AA 80	CALL 80AA	Esegui la subroutine 80AA cioè attendi che si pigi un tasto	
007D 007E	FE 10	CP,10	Controlla se è stato pigiato CONTROL-0	
007F 0080	28 09	JR Z,09	Se è stato pigiato CONTROL-0 va alla riga 008A	
0081 0082	FE 11	CP,11	Controlla se è stato pigiato CONTROL-1	
0083 0084	28 0B	JR Z,0B	Se è stato pigiato CONTROL-1 va alla riga 0090	
0085 0086 0087	CD 0C 80	CALL 800C	Esegui la subroutine 800C	
0088 0089	18 E9	JR,E9	Ritorna alla riga 007A	

Riga	Codice	Mnemonico	Descrizione	Commento
008A 008B 008C	CD 55 80	CALL 8055	Esegui la subroutine 8055	Se è stato pigiato CONTROL-0 carica il numero presente sui display nella coppia di registri DE, lo ricopia su HL, poi ci chiede l'indirizzo finale del «listing»; se è stato pigiato CONTROL-1 carica il numero presente sui display nella coppia di registri DE ed inizia ad eseguire il programma di stampa vero e proprio. In pratica i registri HL contengono l'indirizzo iniziale della stampa mentre i registri DE contengono l'indirizzo finale.
008D	EB	EX DE,HL	Scambia il contenuto di DE con HL	
008E 008F	18 E0	JR,E0	Ritorna alla riga 0070	Trasmette alla stampante il codice ASCII corrispondente all'indirizzo della riga di memoria che sta esplorando.
0090 0091 0092	CD 55 80	CALL 8055	Esegui la subroutine 8055	
0093	7C	LD A,H	Carica in A il contenuto di H	
0094 0095 0096	CD D8 00	CALL 00D8	Esegui la subroutine 00D8	
0097	7D	LD A,L	Carica in A il contenuto di L	
0098 0099 009A	CD D8 00	CALL 00D8	Esegui la subroutine 00D8	
009B 009C	3E 20	LD A,20	Carica in A il numero 20 (spazio bianco)	
009D 009E 009F	CD C1 00	CALL 00C1	Esegui la subroutine 00C1	
00A0 00A1 00A2	CD C1 00	CALL 00C1	Esegui la subroutine 00C1	
00A3	7E	LD A,(HL)	Carica in A il numero contenuto nella riga di memoria indicata da HL ed esegui la subroutine 00D8	
00A4 00A5 00A6	CD D8 00	CALL 00D8		
00A7 00A8	3E 0D	LD A,0D	Carica in A il numero 0D	Trasmette alla stampante un codice convenzionale (0D) che la obbliga a stampare i dati forniti in precedenza.
00A9 00AA 00AB	CD C1 00	CALL 00C1	Esegui la subroutine 00C1	
00AC	7C	LD A,H	Carica in A il contenuto di H	Controlla se ha già listato tutte le righe da noi richieste ed in caso affermativo torna alla riga 0070; se no va a leggere la riga di memoria immediatamente successiva a quella che ha appena stampato.
00AD	BA	CP D	Confronta il contenuto di A con il contenuto di D	
00AE 00AF	20 04	JR NZ,04	Se non sono uguali va alla riga 00B4	
00B0	7D	LD A,L	Carica in A il contenuto di L	
00B1	BB	CP E	Confronta il contenuto di A con il contenuto di E	
00B2 00B3	28 BC	JR Z,BC	Se sono uguali ritorna alla riga 0070	
00B4	23	INC HL	Incrementa il contenuto di HL	
00B5 00B6	18 DC	JR,DC	Ritorna alla riga 0093	
00B7 00B8	D6 0A	SUB 0A	Sottrai al contenuto di A il numero 0A	
00B9 00BA	30 04	JR NC,04	Se il contenuto di A è ancora maggiore o uguale a 0 va alla riga 00BF	

Riga	Codice	Mnemonico	Descrizione	Commento
00BB 00BC	C6 3A	ADD 3A	Addizione al contenuto di A il numero 3A	Converte il codice binario contenuto nel registro A in un codice ASCII in modo da poter applicare questo codice in ingresso alla stampante.
008D 00BE	18 02	JR, 02	Salta alla riga 00C1	
00BF 00C0	C6 41	ADD 41	Addiziona al contenuto di A il numero 41	
00C1	F5	PUSH AF	Salva i contenuti dei registri AF nello Stack	Subroutine di trasmissione dati dalla CPU verso la stampante. Per prima cosa si controlla che il BUSY non sia in condizione logica 1, cioè che la stampante non sia già impegnata a scrivere. Se la stampante è libera si manda in uscita il dato ponendogli artificialmente il bit 7 (strobe) in condizione logica 1. Subito dopo si azzerza per qualche microsecondo il bit 7 in modo da far capire alla stampante che il dato presente in uscita è un dato valido. Si riporta ancora il bit 7 in condizione logica 1 ed a questo punto la stampante accetta in ingresso il dato da noi fornito.
00C2 00C3	DB 03	IN A,(03)	Leggi dalla periferica 03 il numero da caricare in A	
00C4 00C5	CB 47	BIT 0,A	Controlla se il bit 0 di questo numero è uguale a 1 o a 0	
00C6 00C7	20 FA	JR NZ,FA	Se è uguale a 1 ritorna alla riga 00C2	
00C8	F1	POP AF	Riprendi AF nello Stack	
00C9	F5	PUSH AF	Salva nuovamente AF nello Stack	
00CA 00CB	CB FF	SET 7,A	Poni il bit 7 di A in condizione 1	
00CC 00CD	D3 03	OUT(03),A	Trasmetti il contenuto di A alla stampante	
00CE 00CF	CB BF	RES 7,A	Poni il bit 7 di A in condizione 0	
00D0 00D1	D3 03	OUT(03),A	Trasmetti il contenuto di A alla stampante	
00D2 00D3	CB FF	SET 7,A	Poni il bit 7 di A in condizione 1	
00D4 00D5	D3 03	OUT(03),A	Trasmetti il contenuto di A alla stampante	
00D6	F1	POP AF	Riprendi AF dallo Stack	
00D7	C9	RET	Ritorna ad eseguire il programma principale	
00D8	F5	PUSH AD	Salva AF nello Stack	
00D9 00DA	E6 F0	AND F0	Azzerza i bit 0-1-2-3 del registro A	
00DB 00DC 00DD 00DE	0F 0F 0F 0F	RRCA x 4	Shifta di 4 posizioni verso destra i bit del numero contenuto in A	
00DF 00E0 00E1	CD B7 00	CALL 00B7	Esegui la subroutine 00B7	Subroutine per separare la cifra di sinistra da quella di destra in un numero a due cifre in modo da poterle trasmettere separatamente alla stampante la quale ovviamente può stamparne una sola per volta.
00E2	F1	POP AF	Riprendi AF dallo Stack	
00E3	F5	PUSH AF	Salva AF nello Stack	
00E4 00E5	E6 0F	AND 0F	Azzerza i bit 4-5-6-7 del registro A	
00E6 00E7 00E8	CD B7 00	CALL 00B7	Esegui la subroutine 00B7	
00E9	F1	POP AF	Riprendi AF dallo Stack	
00EA	C9	RET	Ritorna al programma principale	