

P6060

**I/O con periferiche esterne
Manuale del programmatore**

olivetti

GP Code 3973710 E (3)

PREFAZIONE

Questa pubblicazione, che illustra la Gestione delle Unità Periferiche e dei Canali di I/O del P6060, è un supplemento al manuale "P6060 Personal Minicomputer - Manuale Generale" ed al manuale "P6060 Extended System Environment - Manuale Generale".

Esso è destinato agli utenti dei minicomputer Olivetti P6060 che necessitano di informazioni dettagliate sulla programmazione delle periferiche esterne.

E' inteso che l'utente deve conoscere il manuale "P6060 Personal Minicomputer - Manuale Generale", oppure il manuale "P6060 Extended System Environment - Manuale Generale", ed i formati dei comandi di sistema e dei verbi BASIC.

Riferimenti:

"P6060 Personal Minicomputer - Manuale generale"
(GR Code 3940910 P (1))

"P6060 Extended System Environment - Manuale generale"
(GR Code 3974520 Y (0))

Distribuzione: Generale (G)

Prima Edizione: Marzo 1977

Seconda Edizione: Ottobre 1977

Terza Edizione : Giugno 1979

PUBBLICAZIONE EMESSA DA:

Ing. C. Olivetti & C., S.p.A.
Direzione Marketing Centrale
Servizio Documentazione
77, Via Jervis - 10015 IVREA (Italy)

STATO DI AGGIORNAMENTO

LIVELLO	DATA	PAGINE AGGIORNATE	PAGINE	CODICE
0	77-03-30		78	3973710 E (0)
1	77-10-30	Edizione definitiva con modifiche	158	3973710 E (1)
2	78-03-15	Prefazione iii iv vi vii 3-1 3-3 5-1 ÷ 5-10 5-12 ÷ 5-20 5-22 5-43 5-59 5-60 Appendice B	58	397312 K
3	79-06-15	Riedizione completa con le pagine seguenti modificate: Prefazione iii + xii 1-7 ÷ 1-16 2-1 ÷ 2-24 3-1 3-3 ÷ 3-11 4-1 4-3 4-4 5-1 5-2 5-22 5-41 5-43 5-47 ÷ 5-52 5-59 ÷ 5-62 Capitolo 6 A-2 A-3 B-22 B-25 ÷ B-29 C-1	134	3973710 E (3)
4				
5				
6				
7				
8				
9				

Le pagine contrassegnate * devono essere annullate

INDICE

<u>INTRODUZIONE</u>	xi	Istruzione TEST	2-19
		Istruzione WAIT	2-21
1. <u>PRINCIPI GENERALI DI I/O CON PERIFERICHE ESTERNE</u>	1-1	Funzione di sistema IOC (num-exp)	2-23
<u>Introduzione</u>	1-1	3. <u>IL SISTEMA P6060 E L'INTERFACCIA IPSO</u>	3-1
<u>Componenti logici di un colloquio tra unita centrale ed unita periferica</u>	1-1	<u>Configurazione del sistema P6060 con periferiche IPSO</u>	3-1
Unita periferica	1-2	<u>Canale IPSO ed istruzioni BASIC</u>	3-3
Canale di I/O	1-5	<u>Esempi di programmazione delle periferiche IPSO</u>	3-5
Programma utente	1-6	4. <u>IMPIEGO DELLA TASTIERA INTEGRATA COME PERIFERICA ESTERNA</u>	4-1
<u>Gestione del canale di I/O</u>	1-11	<u>Canale di I/O</u>	4-1
I/O non sovrapposto e gestione automatica degli errori	1-12	<u>Istruzioni BASIC</u>	4-3
I/O sovrapposto e gestione automatica degli errori	1-13	Istruzione BUFFER	4-3
I/O sovrapposto e gestione da programma utente degli errori	1-14	Istruzione CMD	4-3
<u>Interruzione della esecuzione di un programma utente</u>	1-15	Istruzione INTERRUPT ENABLE	4-4
		Istruzione RECEIVE	4-4
2. <u>LE ISTRUZIONI BASIC DI I/O CON PERIFERICHE ESTERNE</u>	2-1	Istruzione SEND	4-6
<u>Introduzione</u>	2-1	Istruzione TEST	4-6
Istruzione BUFFER	2-3	Istruzione WAIT	4-7
Istruzione CMD	2-5	Funzione di sistema IOC (num-exp)	4-7
Istruzione INTERRUPT ENABLE	2-7	<u>Stato della tastiera</u>	4-7
Istruzione RECEIVE	2-11	<u>Esempi di programmazione</u>	4-8
Istruzione SEND	2-15	5. <u>IL SISTEMA P6060 E L'INTERFACCIA EIA RS-232-C E CURRENT LOOP 20mA</u>	5-1

<u>Introduzione</u>	5-1	<u>Predisposizione dei parametri P per il funzionamento in modo asservito</u>	5-30
<u>L'interfaccia EIA RS-232 C</u>	5-3		
<u>L'interfaccia CURRENT LOOP 20 mA</u>	5-4	<u>Predisposizione dei parametri P per il funzionamento in modo free-running</u>	5-33
<u>Il governo SIC 6629</u>	5-4		
<u>Definizione e funzione dei segnali su cavo MODEM</u>	5-6	<u>Definizione della frase</u>	5-36
Segnali di massa	5-6	<u>Istruzioni BASIC per l'impiego del governo SIC 6629</u>	5-43
Segnali dati	5-6		
Segnali di controllo	5-8	Istruzione BUFFER	5-43
<u>Definizione e funzione dei segnali su cavo EIA</u>	5-9	Istruzione CMD	5-44
Segnali di massa	5-11	Istruzione INTERRUPT ENABLE	5-48
Segnali dati	5-11	Istruzione RECEIVE	5-49
Segnali di controllo	5-11	Istruzione SEND	5-50
		Istruzione TEST	5-50
		Istruzione WAIT	5-50
		La funzione IOC(num-exp)	5-50
<u>Definizione e funzione dei segnali nel collegamento P6060 con P6060</u>	5-13	<u>Istruzioni BASIC per l'impiego di periferiche 20 mA current loop compatibili</u>	5-59
<u>Definizione e funzione dei segnali nel collegamento CURRENT LOOP</u>	5-15	Istruzione BUFFER	5-59
<u>Canali di input/output</u>	5-20	Istruzione CMD	5-59
<u>Modi di funzionamento del sistema P6060 con governo SIC 6629</u>	5-22	Istruzione INTERRUPT ENABLE	5-59
<u>Parametri P</u>	5-24	Istruzione RECEIVE	5-60
Parametri P per il funzionamento in modo asservito	5-25	Istruzione SEND	5-60
Parametri P per il funzionamento in modo free-running	5-27	Istruzione TEST	5-60
<u>Predisposizione del modo di funzionamento</u>	5-30	La funzione di sistema IOC (num-exp)	5-60
		<u>Esempio di programmazione</u>	5-61
		<u>Segnalazioni di errore</u>	5-63
		<u>6. IL SISTEMA P6060 E L'INTERFACCIA IEEE 488-1975</u>	6-1
		<u>Introduzione</u>	6-1
		<u>Prerequisiti del lettore</u>	6-3
		<u>L'interfaccia IEEE 488-1975</u>	6-3

Sistemi con nessuna unità di controllo	6-6	Un parallel poll	6-45
		<u>Installazione di un sistema IEEE 488-1975</u>	6-48
Sistemi con una unità di controllo	6-6		
Sistemi con più unità di controllo	6-7	A. <u>MESSAGGI DI ERRORE</u>	A-1
Handshaking con tre segnali	6-7	B. <u>SCHEDE DI RIFERIMENTO PER LA PROGRAMMAZIONE DELLE PERIFERICHE IPSO</u>	B-1
Collegamento delle apparecchiature	6-9	CR 300	B-3
		CTU 1010	B-5
		LN 20	B-9
<u>Il governo PIC6626</u>	6-11	MTU 1400	B-13
Prestazioni del governo PIC6626	6-14	OPR 1830 WAND	B-15
		PN 20	B-17
		PR 1220/PR 1230/PR 1240	B-21
		PR 1350	B-25
<u>Definizione e funzione dei segnali sul cavo IEEE 488</u>	6-16	C. <u>INTRODUZIONE DA TASTIERA DEI PRIMI 32 CODICI ISO</u>	C-1
<u>Canale di input/output</u>	6-20		
<u>Istruzioni BASIC per l'impiego del governo PIC6626</u>	6-20		
Istruzione BUFFER	6-20		
Istruzione CMD	6-21		
Istruzione INTERRUPT ENABLE	6-25		
Istruzione SEND	6-25		
Istruzione RECEIVE	6-33		
Istruzione TEST	6-35		
Istruzione WAIT	6-36		
Funzione di sistema IOC (num-exp)	6-36		
<u>La programmazione del P6060/PIC6626</u>	6-37		
Programmazione del canale di I/O e del bus	6-38		
Apparecchiature controllate da lontano	6-40		
Un serial poll	6-43		

INDICE DELLE FIGURE

	Pag.	
1-1	Componenti logici di un colloquio tra Unità Centrale e Unità Periferica	1-1
1-2	Schema generale di Unità Periferica	1-3
1-3	Registro di 8 bit per la rappresentazione dell'identificatore di una periferica	1-3
1-4	Canale di I/O	1-5
1-5	Il programma utente ed il deposito di stato corrente	1-9
1-6	Schema logico dello scambio di informazioni con unità periferiche non integrate	1-11
3-1	Periferiche collegabili al Sistema P6060 tramite interfaccia IPSO	3-3
4-1	Impiego della tastiera integrata come periferica esterna	4-2
5-1	Configurazione tipica di sistema	5-2
5-2	Schema dell'interfaccia secondo lo standard EIA	5-3
5-3	Schema dell'interfaccia 20 mA current loop	5-4
5-4	Il governo SIC 6629	5-5
5-5	Segnali trasmessi sui fili del cavo RS 232 MODEM	5-7
5-6	Segnali trasmessi sui fili del cavo RS 232 PERIPH	5-10
5-7	Collegamento tra due P6060	5-13
5-8	Segnali scambiati tra due P6060	5-14
5-9	Segnali trasmessi sui fili del cavo CURRENT LOOP	5-15
5-10	Schema elettrico equivalente dei ricevitori e trasmettitori del governo SIC 6629	5-16

5-11	Collegamento parallelo ricevitore SIC 6629 e trasmettitore 20 mA current loop compatibile	5-17
5-12	Collegamento serie ricevitore SIC 6629 e trasmettitore 20 mA current loop compatibile	5-18
5-13	Collegamento parallelo trasmettitore SIC 6629 e ricevitore 20 mA current loop compatibile	5-19
5-14	Collegamento serie trasmettitore SIC 6629 e ricevitore 20 mA current loop compatibile	5-19
5-15	Canali di Input/Output	5-20
5-16	Il sistema P6060 in modo asservito	5-23
5-17	Il sistema P6060 in free-running	5-23
6-1	Un sistema d'interfaccia standard IEEE 488-1975	6-2
6-2	Sistema senza unità di controllo	6-6
6-3	Sistema con una unità di controllo	6-7
6-4	Sistema con più unità di controllo	6-7
6-5	Esempi di reti	6-10
6-6	Le cinque sezioni logiche del PIC6626	6-13

INDICE DELLE TABELLE

	Pag.
2-1	Informazioni che la funzione IOC (num-exp) può fornire al programma 2-24
4-1	Codici di comando per l'impiego della tastiera integrata come periferica esterna 4-4
4-2	Stato della tastiera 4-7
5-1	Valori di default per i parametri P con P6060 in asservito 5-25
5-2	Valori assegnabili ai parametri P con P6060 in asservito 5-26
5-3	Valori di default per i parametri P con P6060 in free-running 5-28
5-4	Valori assegnabili ai parametri P con P6060 in free-running 5-30
5-5	Tabella di definizione della frase, assunta per default 5-39
5-6	Codici di comando riferibili al trasmettitore o ricevitore 5-45
5-7	Codice di comando per il ricevitore 5-46
5-8	Codici di comando riferibili al trasmettitore con P6060 collegato a DCE 5-47
5-9	Codici di comando riferibili al trasmettitore con P6060 collegato a DTE 5-48
5-10	Informazioni che la funzione IOC (num-exp) può fornire al programma quando il sistema funziona in asservito con cavo RS-232 MODEM 5-52
5-11	Informazioni che la funzione IOC (num-exp) può fornire al programma quando il sistema funziona in asservito con cavo RS-232 PERIPH 5-54
5-12	Informazioni, riferite al canale di input, che la funzione IOC (num-exp) può fornire al programma quando il sistema funziona in free-running con cavo RS-232 MODEM 5-56
5-13	Informazioni, riferite al canale di input, che la funzione IOC (num-exp) può fornire al programma quando il sistema funziona in free-running con cavo RS-232 PERIPH 5-58

6-1	Disposizione delle linee sul connettore	6-16
6-2	Codici di comando per la sezione di INPUT dell'unità di controllo (per-id=16)	6-23
6-3	Codici di comando per la sezione di parlatore (per-id=25)	6-24
6-4	Codici di comando per la sezione di ascoltatore (per-id=17)	6-25
6-5	Messaggi multilinea	6-30
6-6	Caratteri di configurazione del parallel poll	6-31
6-7	Informazioni che la funzione IOC(num-exp) può fornire al programma	6-37
6-8	Caratteri e codici di indirizzo	6-49

INTRODUZIONE

Il Sistema P6060 può essere collegato ad un gran numero di Unità Periferiche scelte tra un'ampia gamma di tipi diversi.

Tale possibilità è attuata grazie all'impiego di diversi canali di I/O ... fino a 16.

E' quindi fondamentale capire in cosa consiste e come può essere gestito un canale di I/O del Sistema P6060: questo è infatti lo scopo del primo capitolo.

Nel secondo capitolo sono descritte, in ordine alfabetico, le istruzioni BASIC che permettono di programmare il colloquio tra l'Unità Centrale e le Unità Periferiche Esterne.

Il terzo capitolo riprende quanto definito nei primi due per personalizzare il discorso all'impiego del canale IPSO.

Il quarto capitolo descrive l'impiego delle istruzioni BASIC di I/O che permettono di programmare l'utilizzazione della tastiera integrata senza fermare l'esecuzione del programma.

Il quinto capitolo descrive l'impiego delle istruzioni BASIC per programmare il colloquio tra P6060 e periferiche con interfaccia seriale coerente con lo standard EIA RS-232-C (CC ITT V 24). In particolare è descritta la possibilità di impiego del sistema P6060 per la trasmissione di dati in time-sharing ad un elaboratore remoto.

Il sesto capitolo descrive come si possa controllare un sistema di misura che utilizza fino a 14 strumenti. Tali strumenti possono essere collegati al P6060 con un governo PIC 6626 mediante un bus d'interfaccia standard IEEE 488-1975.

L'appendice A elenca e descrive gli errori che posso-

no essere segnalati dal Sistema quando si utilizzano delle Unità Periferiche Esterne.

L'Appendice B è una raccolta di schede che riportano per ogni Unità Periferica, con interfaccia IPSO, i codici relativi agli operandi delle diverse istruzioni BASIC che possono essere utilizzate per programmare il colloquio con l'Unità Centrale.

Infine l'Appendice C riporta la corrispondenza tra i primi 32 codici della tabella ISO ed i tasti della tastiera integrata che devono essere premuti insieme al tasto **CONTROL** per introdurli nel relativo buffer.

1. PRINCIPI GENERALI DI I/O CON PERIFERICHE ESTERNE

Introduzione

In questo capitolo sono definiti i componenti logici (programma utente, canale I/O ed unità periferica) che permettono di realizzare lo scambio di informazioni tra l'Unità Centrale del Sistema P6060 ed una generica unità periferica esterna. Successivamente sono descritte le funzioni eseguite da ogni componente per l'attuazione del colloquio suddetto. La descrizione è del tutto generale, infatti sono descritte le funzioni comuni ai diversi tipi di canali I/O utilizzabili sul Sistema P6060, ed è diretta a comunicare al lettore i concetti fondamentali per capire l'impiego più appropriato delle istruzioni BASIC per il colloquio con le unità periferiche esterne.

Componenti logici di un colloquio tra unità centrale ed unità periferica

Quando si comanda all'Unità Centrale di eseguire uno scambio di informazioni o di comandi con una unità periferica, i componenti fondamentali che intervengono nel "colloquio" sono (vedi figura 1-1):

- il programma utente
- il canale I/O
- l'unità periferica

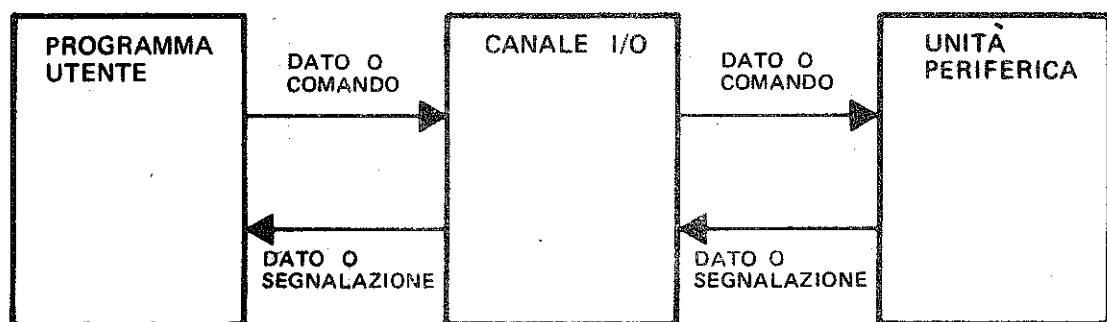


Figura 1-1 Componenti logici di un colloquio tra Unità Centrale ed Unità Periferica

Ogni componente rappresentato in figura esegue un certo numero di funzioni di cui elenchiamo nel seguito solamente quelle fondamentali. Il programma utente comunica al sistema, mediante le istruzioni BASIC descritte nel paragrafo successivo, le operazioni di I/O che devono essere eseguite e con quale sequenza e modalità. Il canale di I/O esegue le operazioni di I/O richieste dal programma, nelle sequenze e modalità specificate. L'Unità periferica riceve i dati trasmessi attraverso il canale I/O, oppure trasmette verso l'Unità Centrale i dati richiesti dal programma e comunica all'Unità Centrale il suo STATO e se i dati ricevuti sono corretti.

Nei paragrafi successivi analizzeremo i componenti logici suddetti iniziando dall'unità periferica, proseguendo con il programma utente e quindi con il canale I/O. Questo ordine di esposizione ci permetterà, infatti, una presentazione coerente delle definizioni che via via scriveremo.

Unità periferica

Da un punto di vista logico, e nel caso più generale, una unità periferica è composta di tre sezioni come descritto in fig. 1-2. I termini OUTPUT ed INPUT fanno riferimento all'Unità Centrale: la sezione di OUTPUT è interessata da una operazione di OUTPUT da Unità Centrale, come il trasferimento di caratteri dall'Unità Centrale al supporto dell'unità periferica; la sezione di INPUT è interessata da una operazione di INPUT verso Unità Centrale, come il trasferimento di caratteri dal supporto dell'unità periferica all'Unità Centrale.

L'unità periferica può anche ricevere comandi dall'Unità Centrale per la sezione di controllo (ad esempio

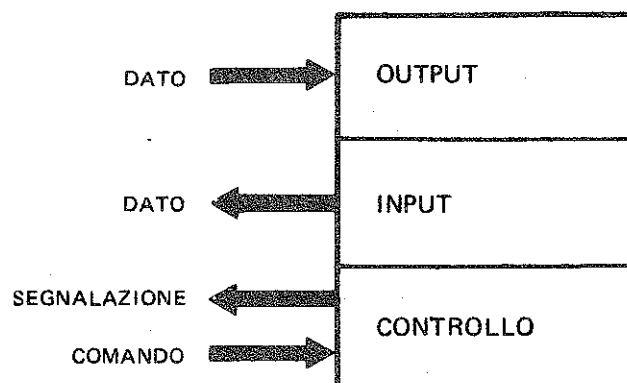


Figura 1-2 Schema generale di Unità periferica

comandi di predisposizione o di sondaggio della posizione del supporto) ed inviare, dalla sezione di controllo, delle segnalazioni per informare l'Unità Centrale del suo stato o dell'esito della operazione richiesta da Unità Centrale.

Le istruzioni di programma, come vedremo nel prossimo paragrafo, specificano con il primo operando, per-id, il canale e la periferica interessati dall'esecuzione dell'istruzione: ossia l'identificatore della periferica. L'identificatore di una periferica è un numero intero che esprime nel sistema decimale il valore di un numero binario di otto cifre rappresentato in un registro di 8 bit. Vediamo nel disegno seguente il significato assegnato dal Sistema alle diverse cifre del numero binario suddetto per identificare il canale I/O, la periferica e definire se la periferica è di INPUT o di OUTPUT.

Bit:	7	6	5	4	3	2	1	0
Peso decimale totale:	128	64	32	16	8	4	2	1
Peso decimale del campo:	8	4	2	1	8	4	2	1
Nome del campo:	Numero di canale				I/O	Numero di sezione di periferica		

0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Figura 1-3 Registro di 8 bit per la rappresentazione dell'identificatore di una periferica

Il significato dei tre campi in cui è diviso il registro di figura 1-3 è il seguente:

1. NOME LOGICO della periferica: è dato dalla configurazione dei primi 3 bit e permette di identificare fino a otto periferiche. Nel caso di unità periferiche con due supporti come le unità per cassetta magnetica CTU 1000 e CTU 1010 si assegnano due nomi logici di periferica: ognuno identificante uno dei due trascinatori.

2. I/O: il valore del quarto bit specifica se la periferica è di INPUT (valore 0) o di OUTPUT (valore 1). Nel caso più generale schematizzato in figura 1-1, si possono utilizzare periferiche che sono contemporaneamente di INPUT e di OUTPUT. Ad esempio l'unità CTU 1000 (unità per cassetta di nastro magnetico) ha una sezione di INPUT ed una di OUTPUT; nel caso in cui vi siano due trascinatori, ad ogni nome logico di periferica corrispondono due nomi per le due sezioni di INPUT e due nomi per le due sezioni di OUTPUT: in tutto quattro sezioni, quindi quattro nomi.
3. NOME LOGICO DI CANALE: individua uno tra 16 diversi canali per ognuna delle 16 configurazioni binarie possibili.

Da quanto detto si deduce che si possono individuare fino a 256 sezioni di unità periferiche distribuite su 16 canali I/O (da zero a 15). Per ogni canale I/O si possono identificare fino a 16 sezioni di unità periferica: 8 di INPUT e 8 di OUTPUT. Così le sezioni di unità periferica collegate con il canale zero potranno essere specificate con identificatori i cui valori sono numeri interi da 0 a 15, quelle collegate con il canale 1 avranno identificatori da 16 a 31 e così via fino a quelle collegate con il canale 15 i cui identificatori varieranno da 240 a 255.

Si noti infine che alcune periferiche sono dotate di un registro di transito su cui sono trasferite le informazioni trasmesse dall'Unità Centrale prima di essere registrate sul supporto (ad esempio nastro magnetico) o su cui sono trasferite dal supporto prima di essere trasmesse all'Unità Centrale. Tale registro è detto buffer e permette di liberare il canale I/O prima che l'informazione ricevuta sia registrata sul supporto esterno o di mantenerlo libero mentre l'informazione è letta dal supporto esterno per essere poi trasmessa all'Unità Centrale. Questo rende le operazioni di I/O indipendenti dai tempi necessari alla periferica per eseguire le sue operazioni e permette una utilizzazione più razionale del canale di I/O.

Il sistema P6060 può utilizzare fino a 16 canali di I/O per il colloquio con unità periferiche esterne.

Un canale di I/O è la somma di più componenti hardware (parte della memoria principale, circuiti logici di Unità Centrale e adattatore di interfaccia con un certo numero di unità periferiche esterne) che sono gestiti da componenti software e firmware per attuare lo scambio di informazioni tra Unità Centrale ed Unità Periferiche.

Il canale di I/O (vedi figura 1-4) è composto da:

- una memoria di transito (BUFFER)
- 16 depositi di stato
- un CONTROLLO CANALE

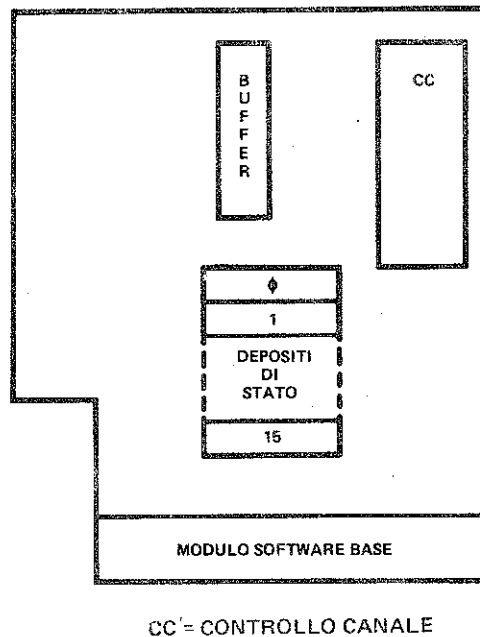


Figura 1-4 Canale di I/O

1. BUFFER

è una parte della memoria principale, la cui dimensione è specificata in una istruzione di programma utente (vedi istruzione BUFFER nel capitolo 2), assegnata al canale I/O per lo scambio di dati con

una unità periferica esterna.

2. DEPOSITO DI STATO

è costituito da due byte (16 bit) di memoria principale: in esso il CONTROLLO DI CANALE registra in formazioni sullo stato del canale, della periferica, sul risultato della trasmissione e sulla interpretazione di alcuni codici ricevuti. Ogni canale I/O contiene 16 depositi di stato ognuno associato ad una sezione di periferica (di INPUT o di OUTPUT).

3. CONTROLLO CANALE

è costituito da componenti firmware ed hardware che, attivati dal modulo di software base, attuano le operazioni richieste.

Un canale I/O è una via indipendente di INPUT/OUTPUT nel senso che le operazioni che si svolgono in un canale non interferiscono con le operazioni che si svolgono sugli altri canali.

Poichè c'è un buffer in memoria principale per ogni canale di I/O, le operazioni di I/O su di un canale possono essere eseguite contemporaneamente ad operazioni di I/O sugli altri canali, contemporaneamente ad operazioni di I/O riferite alle unità periferiche integrate (stampante, display e Floppy Disk) e contemporaneamente ad operazioni di calcolo interno. Questa caratteristica è detta SOVRAPPOSIZIONE e può essere utilizzata durante l'esecuzione delle istruzioni BASIC di I/O specificando in esse l'opzione AND GO (vedi le istruzioni SEND, RECEIVE e CMD).

Si noti infine che sebbene su di un canale possono essere attivate fino ad 8 periferiche di INPUT ed 8 periferiche di OUTPUT, il colloquio su di un canale si attua sempre tra Unità Centrale ed una sola delle periferiche collegate al canale.

Programma utente

Per realizzare un programma che impiega le Unità Periferiche Esterne si possono utilizzare le seguenti istruzioni BASIC:

BUFFER

Assegna una parte della memoria principale all'impiego da parte del programma come memoria di transito per lo scambio dei dati con la unità periferica.

CMD	Trasmette dei comandi operativi o dei comandi di sondaggio all'unità periferica.
INTERRUPT ENABLE	Abilita l'interruzione dell'esecuzione di un programma utente da parte di una periferica esterna.
RECEIVE	Riceve, attraverso il buffer del relativo canale di I/O, il dato presente nel buffer o nel supporto della unità periferica (nel caso in cui manchi il buffer).
SEND	Trasmette, attraverso il buffer del relativo canale di I/O, una stringa di caratteri alla unità periferica specificata nella istruzione.
TEST	Trasferisce nel Deposito di Stato Corrente (vedi definizione più avanti) il contenuto del deposito di stato della periferica specificata nella istruzione.
WAIT	Al termine della esecuzione delle operazioni di I/O in atto sul canale di I/O, trasferisce il contenuto del deposito di stato della periferica specificata con l'istruzione nel deposito di stato corrente.
Funzione IOC (num-exp)	E' una funzione interna di sistema che permette di interrogare il Deposito di Stato Corrente da programma, invece che da sistema operativo.

Nel successivo capitolo le suddette istruzioni sono descritte in dettaglio, per ora ne definiamo le caratteristiche principali per poter capire le modalità di gestione del canale di I/O.

L'istruzione BUFFER è di tipo dichiarativo e quindi può essere posta in qualunque punto del programma; essa assegna un buffer in memoria principale per il canale I/O nel suo complesso; quindi se nel programma vi sono diverse istruzioni BUFFER che specificano diversi identificatori di periferica appartenenti allo stesso canale I/O (per il canale zero, ad esempio, gli identificatori da zero a 15) al canale viene assegnato un buffer con la dimensione più grande tra quelle specificate.

Si noti che la capacità del buffer deve essere dimensionata con riferimento al massimo record che si vuol scambiare con le periferiche collegate al canale di I/O.

L'istruzione `CMD` permette di trasmettere dei comandi di servizio tipici della periferica (non tutte le periferiche hanno comandi di servizio) le cui funzioni e codici dipendono dalla periferica. Quando richiesto dalla particolare periferica, con l'istruzione `CMD` si possono trasmettere anche dei comandi di sondaggio che restituiscono all'Unità Centrale condizioni particolari dell'Unità Periferica.

L'istruzione `INTERRUPT ENABLE` permette ad un programma utente di richiamare una funzione multilinea se si verifica un'interruzione da parte di una periferica esterna mentre il programma è eseguito. In un programma si può avere più di una istruzione `INTERRUPT ENABLE` per richiamare diverse funzioni multilinea per diverse sorgenti d'interruzione. Si possono stabilire diverse priorità entro un insieme di periferiche esterne. Oppure l'effetto di un'istruzione `INTERRUPT ENABLE` può essere annullato con l'esecuzione di un'istruzione di `INTERRUPT ENABLE` successiva.

L'istruzione `RECEIVE` permette di ricevere dati dall'Unità Periferica in due modi diversi (vedi paragrafo "Gestione del canale di I/O") a seconda della presenza o meno nell'istruzione dell'opzione `AND GO`. Se l'opzione `AND GO` non è specificata il dato è ricevuto nel buffer della memoria principale e da qui trasferito nella variabile stringa specificata nella istruzione (vedi capitolo 2). Se è specificata l'opzione `AND GO` il dato è ricevuto nel buffer della memoria principale ma è trasferito nella variabile stringa specificata nella istruzione solo quando è eseguita una successiva istruzione riferita allo stesso canale di I/O.

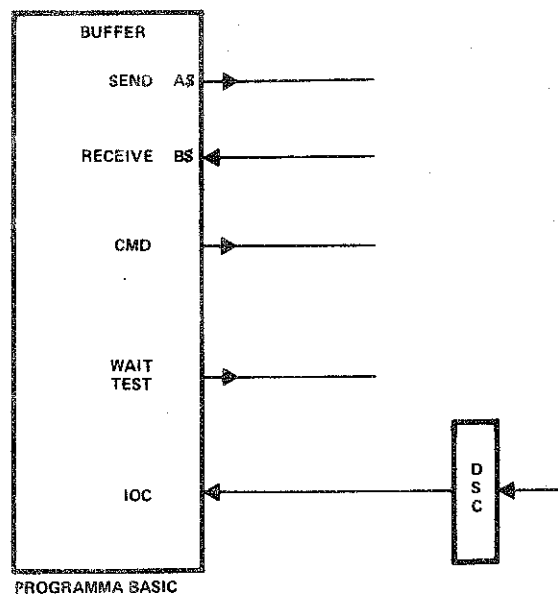
L'istruzione `SEND` permette di trasmettere dati alla Unità Periferica in due modi diversi dipendenti dalla presenza o meno dell'opzione `AND GO` nell'istruzione.

Se non è specificata l'opzione `AND GO` il valore della espressione stringa specificata nella istruzione (vedi capitolo 2.) è trasferito nel buffer della memoria principale e quindi trasmesso all'Unità Periferica. Se l'opzione `AND GO` è specificata, il valore suddetto è trasferito nel buffer della memoria principale e viene da qui trasmesso alla periferica; quando è eseguita una successiva istruzione riferita allo stesso canale di I/O, il dato è sicuramente ricevuto dalla periferica.

Le istruzioni TEST, WAIT e la funzione di sistema IOC (num-exp) permettono di gestire da programma utente le segnalazioni fornite nel Deposito di Stato della relativa periferica.

La presenza di una istruzione TEST o WAIT comunica infatti al Sistema di trasferire nel Deposito di Stato Corrente, vedi figura 1-5, il contenuto del Deposito di Stato della periferica specificata con lo operando per-id nelle istruzioni suddette.

Il sistema dispone di un solo Deposito di Stato Corrente utilizzato da tutti i canali I/O per periferiche esterne; per cui ogni volta che è eseguita una istruzione TEST o WAIT riferita ad una particolare sezione di periferica esterna collegata ad un particolare canale, il precedente contenuto del Deposito di Stato Corrente è sostituito con il contenuto del Deposito di Stato associato a quest'ultima.



DSC = DEPOSITO DI STATO CORRENTE

Figura 1-5 Il programma utente ed il deposito di stato corrente

La funzione di sistema IOC (num-exp), utilizzata in

una istruzione BASIC, permette di gestire da programma le segnalazioni trasferite nel Deposito di Stato Corrente.

Nel caso in cui l'argomento della funzione suddetta è una espressione numerica, questa viene calcolata ed il valore ottenuto, arrotondato all'intero più prossimo, specifica quale delle informazioni contenute nel Deposito di Stato Corrente è testata.

Le seguenti istruzioni sono anche utili nella programmazione delle periferiche:

ASSIGN	Trascodifica una stringa di caratteri ISO nel formato interno e l'assegna ad una o più variabili di programma.
BASSIGN	Assegna ad una o più variabili di programma le stringhe e/o i dati numerici, in formato interno, compresi nel valore di una espressione stringa.
BBUILD	Trasferisce il valore di una o più variabili di programma ad una variabile stringa, mantenendo per essi il formato interno.
BPAD	Eguaglia la lunghezza attuale di una variabile stringa alla sua lunghezza di allocazione, aggiungendo in coda dei caratteri binari.
BUILD	Trascodifica i valori di una o più espressioni in altrettante stringhe ISO e le trasferisce in una variabile stringa.
BUILD USING	Trascodifica i valori di una o più espressioni in altrettante stringhe ISO e le trasferisce in una variabile stringa ponendo i caratteri in posizioni predefinite da una istruzione immagine.
CONVERT	Converte ogni carattere di una espressione stringa nel corrispondente codice numerico ISO e viceversa.
DEPAD	Rimuove in una variabile stringa i caratteri di riempimento specificati.
PAD	Eguaglia la lunghezza attuale di una variabile stringa alla sua lunghezza di allocazione aggiungendo in coda dei caratteri predefiniti.

Per ulteriori informazioni sulle suddette istruzioni si veda il manuale P6060 - Manuale generale.

Gestione del canale di I/O

Dopo aver descritto separatamente la struttura e le funzioni dell'Unità Periferica, del canale di I/O e del programma in memoria principale, analizziamo ora i diversi modi di gestire il colloquio tra Unità Centrale ed Unità Periferiche tenendo presente la figura 1-6 in cui è riportato lo schema logico dello scambio di informazioni con unità periferiche esterne.

Ognuna delle istruzioni esecutive descritte nel paragrafo precedente carica in memoria principale un appropriato modulo del software di base che attraverso il firmware e l'hardware del CONTROLLO CANALE esegue le funzioni specificate dalla istruzione stessa.

Le istruzioni possono essere eseguite diversamente a seconda della modalità di gestione del canale di I/O che è stata scelta.

Tre modi diversi di gestire il colloquio tra Unità Centrale ed Unità Periferiche Esterne sono:

- I/O non sovrapposto e gestione automatica degli errori
- I/O sovrapposto e gestione automatica degli errori
- I/O sovrapposto e gestione da programma utente degli errori.

Naturalmente si possono utilizzare altri modi di gestione del canale di I/O combinando tra loro i tre casi suddetti.

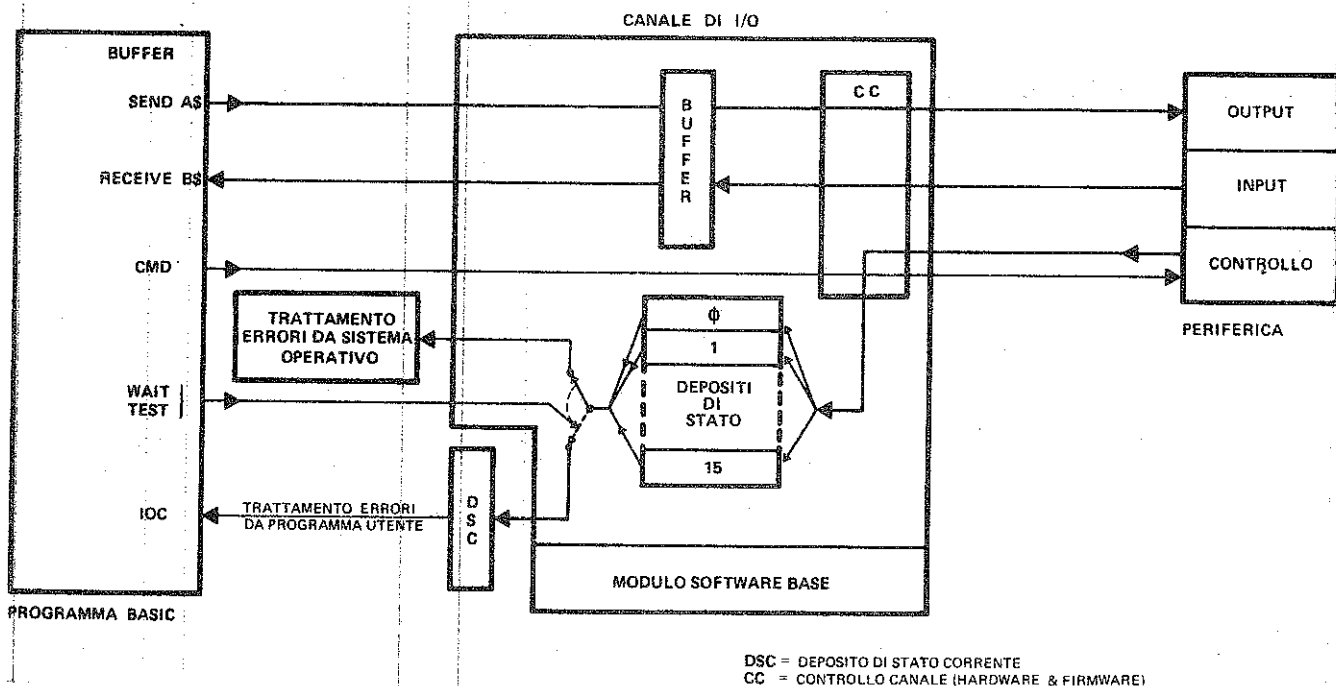


Figura 1-6 Schema logico dello scambio di informazioni con unità periferiche non integrate

I/O non sovrapposto e gestione automatica degli errori

Quando viene scelto questo modo di gestire il colloquio tra l'Unità Centrale e le Unità Periferiche Esterne, le istruzioni BASIC esecutive che lo attuano nel caso più generale di periferiche di INPUT e di OUTPUT sono le seguenti:

- RECEIVE #per-id, string var
- SEND #per-id, string exp
- CMD #per-id, command-code [,command-code] ...

L'istruzione RECEIVE riceve in memoria principale il dato trasmesso dalla Unità Periferica specificata nella istruzione e lo assegna, attraverso il buffer del relativo canale di I/O, alla variabile stringa indicata nella istruzione stessa. Eventuali errori che si verificano durante tali operazioni sono segnalati dal Sistema.

L'istruzione SEND trasmette un dato dall'Unità Centrale ad una Unità Periferica attraverso il buffer del

canale di I/O; eventuali errori che si verificano durante tale scambio di dati sono segnalati dal Sistema.

L'istruzione CMD, infine, trasmette dall'Unità Centrale ad una Unità Periferica un comando o la serie di comandi specificati; eventuali errori che si verificano durante la trasmissione dei suddetti comandi sono segnalati dal Sistema.

Per quanto riguarda la gestione degli errori che si verificano durante lo scambio di informazioni con la periferica, il Sistema Operativo gestisce automaticamente:

- l'errore sullo scambio dati visualizzando ERROR 15 IN LINE
- il FUORI SERVIZIO della Unità Periferica visualizzando ERROR 14 IN LINE
- periferica temporaneamente non disponibile, attendendone la disponibilità.

Riassumendo quindi, il contenuto nella variabile stringa specificata nella istruzione RECEIVE è disponibile appena l'istruzione è eseguita. Il trasferimento nei due sensi del contenuto del buffer del canale di I/O avviene prima che sia terminata l'esecuzione della istruzione. Eventuali errori sono segnalati durante la esecuzione della istruzione.

I/O sovrapposto e gestione automatica degli errori

Quando viene scelto questo modo di gestire il colloquio tra l'Unità Centrale e le Unità Periferiche Esterne, le istruzioni BASIC esecutive che lo attuano nel caso generale di periferiche di INPUT e di OUTPUT sono le seguenti:

- RECEIVE #per-id, string-var AND GO
- SEND #per-id, string-exp AND GO
- CMD #per-id, command-code AND GO

L'istruzione RECEIVE riceve nel buffer del canale di I/O il dato trasmesso dalla Unità Periferica; il dato è assegnato dal buffer suddetto alla variabile stringa specificata nella istruzione quando è eseguita una successiva istruzione di I/O sullo stesso canale. Eventuali errori che si verificano durante lo scambio del dato tra l'Unità Periferica e l'Unità Centrale so-

no segnalati dal Sistema quando è eseguita una successiva istruzione di I/O sulla stessa periferica.

L'istruzione SEND trasferisce nel buffer del canale di I/O il dato da trasmettere ad una Unità Periferica; la trasmissione del dato all'Unità Periferica sarà sicuramente terminata quando una successiva istruzione di I/O sarà eseguita sullo stesso canale. Eventuali errori che si verificano durante lo scambio del dato con l'Unità Periferica sono segnalati dal Sistema quando è eseguita una successiva istruzione di I/O sulla stessa periferica.

L'istruzione CMD trasmette all'Unità Periferica il comando specificato. Eventuali errori di trasmissione sono segnalati quando è eseguita una successiva istruzione di I/O sulla stessa periferica.

Quindi, gli errori che si verificano durante la esecuzione di una generica istruzione di I/O sono segnalati quando inizia l'esecuzione di una successiva istruzione di I/O riferita alla stessa Unità Periferica.

I/O sovrapposto e gestione da programma utente degli errori

Quando viene scelto questo modo di gestire il colloquio tra l'Unità Centrale e le Unità Periferiche Esterne, le istruzioni BASIC esecutive che lo attuano nel caso più generale di periferiche di INPUT e di OUTPUT sono le seguenti:

- RECEIVE #per-id, string-var AND GO
- SEND #per-id, string-exp AND GO
- CMD #per-id, command code AND GO
- WAIT #per-id, oppure TEST #per-id

Le prime tre istruzioni eseguono le funzioni descritte nel paragrafo precedente con le stesse modalità; in fatti avendo specificato in esse l'opzione AND GO esse sono eseguite in sovrapposizione.

Se osserviamo la figura 1-6 vediamo che le istruzioni TEST e WAIT permettono di commutare l'interruttore verso il Deposito di Stato Corrente; questo indica simbolicamente che le informazioni presenti nel Deposito di Stato della periferica specificata nelle istruzioni stesse sono trasferite nel Deposito di Stato Corrente ed il Sistema operativo non produce alcun mes-

saggio nel caso la Unità Periferica sia FUORI SERVIZIO o si verifichi un errore durante lo scambio dati.

Utilizzando le istruzioni WAIT o TEST, le informazioni presenti nel Deposito di Stato Corrente, sondabile mediante la funzione IOC (num-exp), devono essere gestite da programma utente.

Si noti che quando nel programma utente non sono utilizzate le istruzioni TEST o WAIT il Sistema Operativo gestisce automaticamente le situazioni di errore depositate nel Deposito di Stato della Periferica: in figura 1-6 tale situazione è simbolicamente rappresentata con l'interruttore posto verso il lato indicato con TRATTAMENTO ERRORI DA SISTEMA OPERATIVO.

Le istruzioni WAIT o TEST devono essere eseguite prima di eseguire una successiva istruzione di I/O riferita alla stessa periferica altrimenti il Sistema Operativo segnala gli eventuali errori che si sono verificati nella precedente operazione sulla stessa periferica.

L'istruzione WAIT attende che l'operazione di I/O in corso sulla periferica specificata termini, prima di trasferire nel Deposito di Stato Corrente le informazioni contenute nel relativo Deposito di Stato della periferica. L'istruzione TEST, invece, esegue immediatamente tale trasferimento per cui utilizzando da programma utente la funzione di sistema IOC (num-exp) con un argomento appropriato (valore uguale a otto) si possono fare eseguire altre operazioni all'Unità Centrale mentre l'operazione in corso sulla periferica suddetta prosegue fino al suo termine. Questa possibilità diventa particolarmente importante quando in memoria principale risiede un programma che gestisce il colloquio con diversi canali, e si voglia realizzare l'impiego più razionale della Unità Centrale nella gestione del colloquio con le periferiche collegate con i diversi canali.

Interruzione della esecuzione di un programma utente

L'esecuzione di un programma utente può essere interrotta da alcune cause che si riferiscono allo scambio dati con periferiche esterne. In questo caso l'utente deve avere prima digitato il comando OPTIONS con la opzione XNT e nel programma utente deve essere presente un'istruzione di abilitazione dell'interruzione,

istruzione INTERRUPT ENABLE, unitamente ad una definizione di funzione multilinea che viene eseguita quando si verifica la causa d'interruzione.

Nel capitolo successivo è descritta l'istruzione INTERRUPT ENABLE mentre nei capitoli che lo seguono verrà riportata la causa (o le cause) che fanno scattare il meccanismo d'interruzione suddetto per il relativo canale di I/O.

Introduzione

In questo capitolo sono descritte in dettaglio le istruzioni BASIC per il colloquio con le unità periferiche esterne delle quali abbiamo già definito nel capitolo precedente le funzioni.

Le istruzioni sono riportate in ordine alfabetico e per ognuna di esse è definita la funzione, la sintassi e le azioni prodotte quando è eseguita. Seguono alcune note ed un esempio di codifica della istruzione che ha unicamente lo scopo di chiarirne la sintassi.

Alla fine del capitolo viene anche descritta la funzione di sistema IOC (num-exp).

Istruzione BUFFER**Funzione**

Assegna ad un canale di I/O una parte della memoria principale come deposito temporaneo delle informazioni da scambiare con le periferiche esterne.

Formato

BUFFER #per-id, buffer-size

dove:

per-id

è un numero intero compreso tra 0 e 255

buffer-size

è un numero intero compreso tra 1 e 65536 (byte).

Azione

In memoria principale viene assegnata, al canale di I/O a cui è collegata la Unità Periferica di nome per-id, un'area di transito per lo scambio dei dati. La capacità di tale area è specificata in byte dall'operando buffer-size.

Note

1. L'istruzione BUFFER è una istruzione dichiarativa per cui può essere specificata in qualunque punto di un programma.
2. Se in un programma è presente più di una istruzione BUFFER riferita al medesimo canale di I/O (un canale può essere identificato con uno qualunque tra 16 codici, associato ad una delle 16 possibili sezioni di Unità Periferica ad esso collegabili), l'area di transito per lo scambio dei dati che è assegnata al canale di I/O avrà la capacità espressa dal massimo tra i valori degli operandi buffer-size presenti nelle suddette istruzioni.

3. Quando si impiega una istruzione RECEIVE, il valore della lunghezza di allocazione della variabile stringa in cui è trasferito il dato ricevuto dalla periferica non deve essere inferiore alla capacità del buffer assegnato al rispettivo canale di I/O.

Esempio

Nell'esempio seguente al canale di I/O specificato con i codici per-id da 0 a 15, è assegnato un buffer di 256 byte, mentre al canale di I/O specificato con i codici per-id da 16 a 31 è assegnato un buffer di 64 byte.

```
10 BUFFER #2,128
```

```
-
```

```
-
```

```
-
```

```
200 BUFFER #10,256
```

```
210 BUFFER #20,64
```

```
-
```

```
-
```

```
-
```

Istruzione CMD

Funzione

Trasmette ad una Unità Periferica uno o più comandi che essa deve eseguire.

Formato

CMD #per-id, command-code [, command-code] ... [AND GO]

dove:

per-id

è una espressione numerica il cui valore arrotondato all'intero più prossimo deve essere compreso tra 0 e 255

command-code

è un numero intero che specifica un codice di comando

AND GO

specifica che l'ultimo comando è eseguito in sovrapposizione con altre operazioni comandate dall'Unità Centrale.

Azione

1. Se non è specificata l'opzione AND GO, i comandi i dentificati con gli operandi command-code sono trasmessi, con la sequenza con cui sono specificati nella istruzione, all'Unità Periferica identificata con il valore arrotondato all'intero più prossimo della espressione numerica per-id. Gli errori che si verificano durante la trasmissione di un comando all'Unità Periferica sono segnalati dal Sistema. Nel Deposito di Stato associato con tale periferica sono trasferite le informazioni che ne definiscono lo STATO.
2. Se è specificata l'opzione AND GO, i comandi i dentificati con gli operandi command-code sono tra-

smessi come detto in precedenza ma gli errori che si verificano durante la trasmissione dell'ultimo comando specificato prima di AND GO sono visualizzati quando è eseguita una successiva istruzione sulla stessa periferica.

INTERRUPT ENABLE

Istruzione INTERRUPT
ENABLE

Funzione

Abilita l'interruzione dell'esecuzione di un programma utente da parte di una periferica esterna.

Formato

INTERRUPT ENABLE (E,funam,intmsk[,priorit])

dove:

E

è l'omonima lettera che specifica che una periferica esterna sarà in grado d'interrompere l'esecuzione del programma utente.

funam

è una lettera maiuscola dell'alfabeto inglese, o una variabile stringa il cui valore è una lettera maiuscola dell'alfabeto inglese, che corrisponde al nome di una funzione multilinea definita dallo utente.

intmsk

è una stringa di 16 caratteri 0 o 1, o una variabile stringa il cui valore è una stringa di 16 caratteri 0 o 1, che rappresenta la maschera d'interruzione. Ad ogni carattere è associato un canale di I/O esterno. Se consideriamo ogni carattere della stringa come un bit di una maschera a 16 bit, indicando come bit 0 il primo da sinistra e bit 15 il sedicesimo da sinistra, l'associazione tra i caratteri della stringa (bit) ed i canali di I/O è la seguente:

<u>Bit</u>	<u>Canale di I/O</u>
15	(Riservato per impiego interno Olivetti)
14	(Non assegnato)

- 13 (Non assegnato)
- 12 (Non assegnato)
- 11 (Non assegnato)
- 10 Canale 10, RS-232-C o Current Loop 20mA
- 9 Canale 9, RS-232-C o Current Loop 20mA
- 8 Canale 8, RS-232-C o Current Loop 20mA
- 7 Canale 7, RS-232-C o Current Loop 20mA
- 6 Canale 6, RS-232-C o Current Loop 20mA
- 5 Canale 5, RS-232-C o Current Loop 20mA
- 4 Canale 4, RS-232-C o Current Loop 20mA
- 3 Canale 3, RS-232-C o Current Loop 20mA
- 2 Canale 2 per tastiera integrata impiegata come periferica esterna
- 1 Canale 1, IPSO od IEEE 488-1975
- 0 Canale 0, IPSO

priorit

è una variabile numerica contenente un valore da 0 a 15 che stabilisce un livello di priorità tra due o più interruzioni che dovessero verificarsi contemporaneamente.

Azione

L'istruzione comanda l'esecuzione della funzione multilinea definita dall'utente di nome funam quando la periferica collegata al canale specificato con il relativo bit a 1 in intmsk emette una interruzione.

Quando è terminata l'esecuzione della funzione multilinea viene eseguita l'istruzione successiva a quella in cui si è verificata l'interruzione.

Note

1. La prima istruzione della funzione multilinea avrà il formato:

```
DEF FN  $\alpha$  (num-var)
```

dove α è una lettera maiuscola dell'alfabeto inglese che specifica il nome della funzione e num-var è una variabile numerica il cui valore è il numero di linea dell'istruzione in cui è avvenuta l'interruzione del programma utente. Vedi la definizione di funzione multilinea nel capitolo 5 del manuale "P6060 - Manuale generale" nel paragrafo DEF/FNEND.

Quando la funzione multilinea è eseguita deve assegnare a FN* un valore che però non ha alcun significato.

2. Se il valore di funam è asterisco "*" sono disabilitate le interruzioni da parte delle periferiche collegate ai canali per i quali è specificato un 1 nella relativa posizione dell'operando intmsk.
3. Se l'operando priorit non è specificato, oppure è specificato con un operando intmsk che abilita più di una sorgente d'interruzione, è assunta come priorità più alta quella relativa al canale associato con il bit 15 e così via nell'ordine decrescente fino al canale associato con il bit 0.
4. Le cause che per i diversi canali danno luogo ad un'interruzione, se questa è abilitata, sono le seguenti:
 - le periferiche collegate al canale IPS0, 0 o 1, interrompono per fine scambio dati
 - le periferiche collegate all'interfaccia IEEE 488-1975 interrompono per fine scambio dati o perché c'è stata una richiesta di servizio (RQS)
 - la tastiera integrata impiegata come periferica esterna interrompe quando è premuto il tasto END OF LINE in risposta ad un'istruzione RECEIVE
 - infine le apparecchiature collegate al canale RS-232-C o Current Loop 20mA interrompono per fine scambio dati, se funzionano in modo asser-

vito, mentre se funzionano in modo free-running interrompono ogni qualvolta è ricevuto un qualsiasi separatore.

5. Le interruzioni sui vari canali sono abilitate dal momento in cui viene eseguita un'istruzione INTERRUPT ENABLE sino al termine del programma o alla esecuzione di una successiva istruzione INTERRUPT ENABLE. Il risultato della maschera di successive istruzioni INTERRUPT ENABLE è dato dall'OR delle maschere relative.
6. L'istruzione INTERRUPT ENABLE richiede l'esecuzione del comando di sistema OPTIONS con l'opzione XNT.

Istruzione **RECEIVE**

Funzione

Trasferisce da una Unità Periferica alla Unità Centrale un dato in forma di stringa di caratteri pari al massimo alla lunghezza di allocazione della variabile stringa specificata.

Formato

RECEIVE #per-id, string-var [AND GO]

dove:

per-id

è una espressione numerica il cui valore arrotondato all'intero più prossimo, indicando l'identificatore di una sezione di INPUT, deve essere compreso in uno dei seguenti range: 0-7, 16-23, 32-39, 48-55, 64-71, 80-87, 96-103, 112-119, 128-135, 144-151, 160-167, 176-183, 192-199, 208-215, 224-231, 240-247

string-var

è una variabile stringa

AND GO

specifica che l'istruzione è eseguita in sovrapposizione con altre operazioni comandate dall'Unità Centrale.

Azione

1. Se non è specificata l'opzione AND GO, la stringa di caratteri trasmessa all'Unità Centrale dall'Unità Periferica identificata con l'operando per-id è assegnata alla variabile stringa string-var, attraverso il buffer del canale di I/O a cui è collegata la suddetta periferica. Gli eventuali errori che si verificano durante la trasmissione del dato all'Unità Centrale sono segnalati dal Sistema.
2. Se è specificata l'opzione AND GO, la stringa di caratteri trasmessa all'Unità Centrale dalla Unità Periferica

riferica identificata con l'operando per-id è ricevuta nel buffer del canale di I/O a cui è collegata la suddetta periferica.

Il trasferimento del dato nella variabile stringa specificata con l'operando string-var avviene quando è eseguita una successiva istruzione sullo stesso canale.

Eventuali errori che si verificano durante la trasmissione del dato sono visualizzati dal Sistema quando è eseguita una successiva istruzione sulla stessa periferica.

Note

1. Se l'Unità Periferica specificata con per-id è dotata di buffer, l'istruzione RECEIVE deve essere preceduta da una istruzione CMD che trasmetta all'Unità Periferica il comando operativo che trasferisce il dato dal supporto esterno al buffer suddetto.
2. La lunghezza di allocazione della variabile stringa string-var non deve essere superiore alla capacità del buffer del canale di I/O a cui la periferica identificata con per-id è collegata.
3. Se sono ricevuti meno caratteri di quelli determinati dalla lunghezza di allocazione della variabile stringa specificata nell'istruzione RECEIVE la funzione IOC (5) ritorna il valore uno se è eseguita un'istruzione TEST o WAIT riferita alla stessa periferica.

Esempi

1. Vediamo un esempio di impiego delle istruzioni RECEIVE eseguite in sovrapposizione.

-
-
-

100 RECEIVE #4,B\$ AND GO
110 RECEIVE #4,A\$ AND GO (qui in B\$ c'è il dato)

-

- Elaborazione su B\$

-

200 RECEIVE #4,B\$ AND GO (qui in A\$ c'è il dato)

-

- Elaborazione su A\$

-

2. In questo programma si può utilizzare ogni volta il precedente contenuto di A\$.

-

-

-

100 RECEIVE #4,A\$ (Assegna il primo valore ad A\$)
110 RECEIVE #4,A\$ AND GO

-

- Elaborazione del contenuto precedente di A\$

-

190 IF A=B THEN 220

200 GO TO 110

220 CMD#4,Ø (Fa in modo che ad A\$ sia
assegnato l'ultimo valore)

-

- Elaborazione dell'ultimo valore di A\$

-

Istruzione SEND

Funzione

Trasferisce ad una Unità Periferica un dato in forma di stringa di caratteri.

Formato

SEND #per-id, string-expr [AND GO]

dove:

per-id

è una espressione numerica il cui valore arrotondato all'intero più prossimo, indicando l'identificatore di una sezione di OUTPUT, deve essere compreso in uno dei seguenti range: 8-15, 24-31, 40-47, 56-63, 72-79, 88-95, 104-111, 120-127, 136-143, 152-159, 168-175, 184-191, 200-207, 216-223, 232-239, 248-255

string-expr

è una espressione stringa

AND GO

specifica che l'istruzione è eseguita in sovrapposizione con altre operazioni comandate da Unità Centrale.

Azione

1. Se non è specificata l'opzione AND GO, la stringa di caratteri corrispondente a string-expr è trasmessa all'Unità Periferica identificata con il valore arrotondato all'intero più prossimo della espressione numerica per-id. Gli eventuali errori che si verificano durante la trasmissione del dato sono segnalati dal Sistema.
2. Se è specificata l'opzione AND GO, la stringa di caratteri corrispondente a string-expr è trasferita nel buffer del canale di I/O a cui è collegata la Unità Periferica identificata con l'operando per-id.

La trasmissione del dato alla Unità Periferica suddetta è sicuramente terminata quando è eseguita una successiva istruzione di I/O sullo stesso canale.

Gli eventuali errori che si verificano durante la trasmissione del dato sono visualizzati dal Sistema quando è eseguita la successiva istruzione di I/O sulla stessa periferica.

Note

1. Se l'Unità Periferica specificata con per-id è dotata di buffer, in esso sono registrati i caratteri trasmessi dalla Unità Centrale.

Per trasmettere successivamente i caratteri da tale buffer al supporto esterno si deve trasmettere da Unità Centrale un comando che dipende dal tipo di periferica.

2. Il numero di caratteri (lunghezza attuale) della stringa corrispondente a string-exp non deve essere più grande della capacità del buffer del canale di I/O a cui la periferica specificata con per-id è collegata.

Esempi

1. La routine sottostante mostra come il dato ricevuto dalla periferica 4 è inviato successivamente alla periferica 31 ed entrambe le operazioni sono effettuate in sovrapposizione. Si noti che la prima SEND eseguita trasmette un valore vecchio.

```
100 FOR I = 1 TO N
110 RECEIVE# 4,A$ AND GO
120 SEND# 31,A$ AND GO
130 NEXT I
140 SEND# 31,A$
```

2. Nella seguente routine il calcolo della espressione stringa nella istruzione 110 è eseguito prima che in A\$ sia assegnato il suo contenuto dalla istruzione 100, eseguita in sovrapposizione. Quindi l'istruzione 110 trasmette alla periferica la stringa di caratteri "***OLIVETTI P6060***" e non la stringa di caratteri che si ottiene aggiungendo al nuovo valore di A\$ i tre asterischi.

-
-
-
90 A\$ = "****OLIVETTI P6060"
100 RECEIVE # 4,A\$ AND GO
110 SEND # 12,A\$ + "****"
-
-
-

Istruzione TEST

Funzione

Trasferisce nel Deposito di Stato Corrente le informazioni contenute nel Deposito di Stato associato ad una Unità Periferica, senza attendere la fine di una eventuale operazione di I/O eventualmente in corso sulla periferica specificata.

Formato

TEST #per-id

dove:

per-id

è una espressione numerica il cui valore arrotondato all'intero più prossimo è compreso tra 0 e 255.

Azione

Le informazioni contenute nel Deposito di Stato associato alla periferica identificata dal valore arrotondato all'intero più prossimo dell' espressione numerica per-id sono trasferite nel Deposito di Stato Corrente.

Nel Deposito di Stato associato alla periferica sono azzerati i due bit corrispondenti alla informazione di periferica fuori servizio ed errore di scambio dati: questo impedisce la segnalazione dei corrispondenti errori da parte del Sistema Operativo.

Note

1. L'istruzione TEST deve essere eseguita prima che sia eseguita una successiva istruzione di I/O sulla stessa periferica, altrimenti:

- il sistema gestisce automaticamente gli errori
- se la periferica impegna il canale, il sistema attende che questo si liberi.

2. Per interrogare da programma il contenuto del Deposito di Stato Corrente si deve utilizzare la funzione di Sistema IOC (num-exp).

Istruzione WAIT**Funzione**

Attende che sia terminata l'operazione di I/O sulla periferica specificata e quindi trasferisce nel Deposito di Stato Corrente le informazioni contenute nel Deposito di Stato associato ad una Unità Periferica.

Formato**WAIT #per-id**

dove:

per-id

è una espressione numerica il cui valore arrotondato all'intero più prossimo è compreso tra 0 e 255.

Azione

Le informazioni contenute nel Deposito di Stato associato alla periferica identificata dal valore arrotondato all'intero più prossimo della espressione numerica per-id sono trasferite nel Deposito di Stato Corrente.

Nel Deposito di Stato associato alla periferica sono azzerati i due bit corrispondenti alla informazione di periferica fuori servizio ed errore di scambio dati: ciò impedisce la segnalazione dei corrispondenti errori da parte del Sistema Operativo.

Note

1. L'istruzione WAIT deve essere eseguita prima che sia eseguita una successiva istruzione di I/O sulla stessa periferica altrimenti il sistema gestisce automaticamente gli errori.
2. Per interrogare da programma il contenuto del Deposito di Stato Corrente si deve utilizzare la funzione di Sistema IOC (num-exp).

Funzione di sistema
IOC(num-exp)

Funzione

Permette di verificare il contenuto del Deposito di Stato Corrente dopo che è eseguita una istruzione TEST o WAIT.

Formato

IOC(num-exp)

dove:

num-exp è un'espressione numerica il cui valore è un numero intero compreso tra 1 e 8 e seleziona un bit del Deposito di Stato Corrente. Il significato di tali bit è riportato qui nel seguito sebbene alcuni particolari dipendano dal tipo d'interfaccia utilizzato, vedi i capitoli 3, 4, 5 e 6.

Azione

La funzione IOC(num-exp) assume il valore 0 od 1 del bit selezionato nel Deposito di Stato Corrente.

Note

1. Questa funzione normalmente è usata in una delle seguenti istruzioni BASIC come segue:

```
IF IOC(num-exp) = 1 THEN line-num  
ON IOC(num-exp) GOTO line-num  
ON IOC(num-exp) GO SUB line-num
```

2. In tabella 2-1 sono rappresentate le informazioni che si possono fornire al programma utente mediante l'impegno della funzione IOC(num-exp).

Valore di <u>num-exp</u> arrotondato all' <u>intero</u> più prossimo	Informazione fornita se il valore ritornato da IOC (num-exp) è <u>uno</u>	Informazione fornita se il valore ritornato da IOC (num-exp) è <u>zero</u>
1	La Unità Periferica non è disponibile <u>temporaneamente</u>	La Unità Periferica è disponibile
2	Errore durante lo scambio dati con la Unità Periferica	Nessun errore durante lo scambio dati con la Unità Periferica
3	Informazione che dipende dal tipo di canale di I/O	Informazione che dipende dal tipo di canale di I/O
4	"	"
5	"	"
6	L'Unità Periferica è FUORI SERVIZIO. In questo caso le informazioni precedenti non hanno significato	L'Unità Periferica è IN SERVIZIO
7	Il canale di I/O è occupato	Il canale di I/O è libero
8	L'Unità Periferica è impegnata in un "colloquio" con L'Unità Centrale. In questo caso le informazioni precedenti non hanno significato.	L'Unità Periferica non è impegnata in un "colloquio" con l'Unità Centrale

Nota Se il valore di num-exp arrotondato all'intero più prossimo è diverso da quelli specificati nella prima colonna, allora la funzione IOC (num-exp) ritorna un valore decimale corrispondente ad uno tra 128 codici il cui significato dipende dal tipo di canale di I/O.

Tabella 2-1 Informazioni che la funzione IOC(num-exp) può fornire al programma

3. IL SISTEMA P6060 E L'INTERFACCIA IPSO

Configurazione del sistema P6060 con periferiche IPSO

Al Sistema P6060 si possono collegare le seguenti Unità Periferiche:

PN 20	Perforatore di nastro
LN 20	Lettore di nastro perforato
CTU 1010	Unità nastro a cassetta (ad un trascinatore)
CTU 1050	Secondo trascinatore per CTU 1010 (per collegare il CTU 1050 al CTU 1010 si deve utilizzare il cavo IC 1010)
OPR 1830	Lettore ottico
MTU 1400	Unità nastro magnetico compatibile IBM
PR 1220	Stampante ausiliaria veloce (100 car/sec)
PR 1230	Stampante ausiliaria veloce (175 car/sec)
PR 1240	
PR 1350	Stampante ausiliaria veloce (300 car/sec)
PR 1370	
CR 300	Lettore di schede perforate

L'Olivetti ha inoltre certificato il collegamento con le seguenti periferiche di altre case costruttrici:

FACIT 4070 Perforatore veloce di nastro

GOERZ Servogor Plotter 211 - Tracciatore grafico

BENSON Plotter - Tracciatore grafico

Per poter programmare, con il Sistema P6060, una qualunque delle suddette Unità Periferiche, si deve inserire nella Unità base un adattatore d'interfaccia IPSO 6600. Ad ogni adattatore d'interfaccia IPSO 6600 si possono collegare fino a 4 unità periferiche simili o diverse. Sull'unità base si possono montare fino a due adattatori d'interfaccia IPSO 6600: per cui un Sistema P6060 può utilizzare fino a 8 unità periferiche, scelte tra quelle suddette.

L'adattatore IPSO 6600 fa riferimento, all'interno del Sistema, ad un governo, dei circuiti logici ed una parte di memoria principale impiegata come buffer (la cui dimensione è definibile da programma: vedi istruzione BUFFER); l'insieme delle parti suddette viene definito canale IPSO e permette la comunicazione di comandi ed informazioni tra il Sistema ed una periferica IPSO.

In figura 3-1 sono indicate graficamente le unità periferiche IPSO collegabili al Sistema P6060.

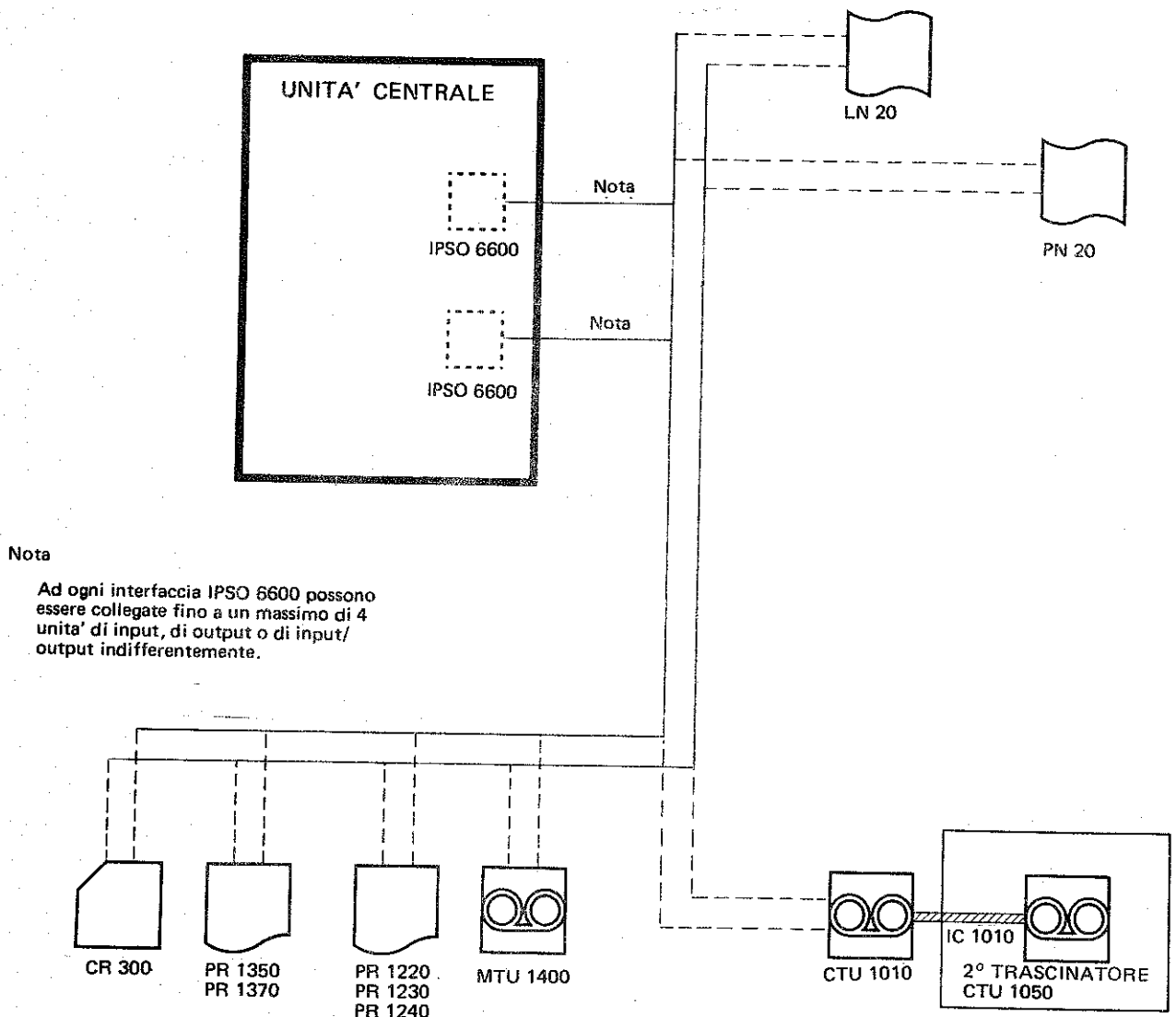


Figura 3-1 Periferiche collegabili al sistema P6060 tramite interfaccia IPSO

Canale IPSO ed istruzioni BASIC

Per quanto riguarda la gestione del colloquio tra Unità Centrale e le periferiche elencate nel paragrafo precedente valgono le modalità descritte nel capitolo 1.

Poiché nel caso dei due canali IPSO i nomi logici utilizzabili sul sistema P6060 sono zero ed uno, i nomi

che si potranno assegnare alle 4 sezioni di INPUT e 4 sezioni di OUTPUT dei due canali sono:

CANALE 0		CANALE 1	
UNITA' DI INPUT	UNITA' DI OUTPUT	UNITA' DI INPUT	UNITA' DI OUTPUT
0 ÷ 7	8 ÷ 15	16 ÷ 23	24 ÷ 31

Nella Appendice B sono riportati i nomi definiti come standard per le periferiche IPSO.

Le istruzioni BASIC che permettono di programmare il colloquio tra Unità Centrale e queste Unità Periferiche Esterne sono:

```
BUFFER    per-id, buffer-size
CMD #    per-id, command-code, [command-code]...[AND GO]
INTERRUPT ENABLE(E, funam, intmask[, priort])
RECEIVE # per-id, string-var[AND GO]
SEND #    per-id, string-exp[AND GO]
TEST #    per-id
WAIT #    per-id
```

L'operando per-id è sostituito con il nome della periferica. Nella istruzione SEND per-id potrà essere compreso tra 8 e 15 o tra 24 e 31 a seconda che la periferica sia collegata al canale 0 o al canale 1. Nella istruzione RECEIVE per-id potrà essere compreso tra 0 e 7 o tra 16 e 23 rispettivamente per il canale 0 o per il canale 1. Nelle istruzioni BUFFER, TEST e WAIT per-id potrà essere compreso tra 0 e 15 oppure tra 16 e 31 rispettivamente con riferimento al canale 0 oppure al canale 1. Infine per quanto riguarda l'istruzione CMD tanto per l'operando per-id quanto per gli operandi command-code si vedano le schede riportate nell'Appendice B.

Per quanto riguarda, infine, il significato delle informazioni che si possono sondare nel Deposito di Stato Corrente con l'impiego della funzione IOC (num-exp), si noti che quando il valore ritornato dalla funzione

è uno nei casi non riportati in tabella 1-1 si ha:

Se IOC (3) = 1 richiesta di servizio da parte di una o più periferiche collegate al canale

Se IOC (4) = 1 si è verificata la condizione selezionata con l'ultimo comando trasmesso alla periferica

Se IOC (5) = 1 l'Unità Periferica ha meno caratteri di quelli richiesti da Unità Centrale

Si noti che il valore riportato tra parentesi è quello corrispondente al valore di num-exp arrotondato all'intero più prossimo.

Se il valore di num-exp, arrotondato all'intero più prossimo, non è compreso tra uno e otto, il valore ritornato dalla suddetta funzione è zero ma non è significativo.

Nota: Se una stampante collegata ad un canale IPSO è comandata da un programma mediante istruzioni PRINT (vedi comando CONFIGURE nel manuale P6060 - Personal Minicomputer Manuale Generale), queste non devono essere eseguite tra una istruzione SEND o RECEIVE riferita al canale IPSO suddetto e la relativa istruzione WAIT o TEST.

Esempi di programmazione delle periferiche IPSO

Riportiamo in questo paragrafo alcuni esempi di impiego delle istruzioni BASIC di I/O generale per mostrare come sia facile il loro impiego: si tratta infatti di sostituire in ogni istruzione il valore appropriato per ogni operando, dedotto dalle schede riportate nella Appendice B.

Per quanto riguarda l'impiego dei singoli comandi si vedano i rispettivi manuali di periferica il cui codice e titolo è riportato sempre nelle schede dell'Appendice B.

Esempio 1: La routine sottostante registra 10 record sul nastro magnetico di una cassetta inserita nel CTU 1010.

L'istruzione BUFFER riserva 80 byte di memoria principale per il canale IPSO (canale zero).

L'istruzione CMD trasmette alla Unità Periferica tre comandi operativi in sequenza 7,5 e 8 (rispettivamente inizializza il nastro, registra un blocco di controllo, azzerà il pointer); vedi la relativa scheda per il CTU 1010 nell'Appendice B.

L'istruzione SEND trasmette alla Unità Periferica il record di caratteri introdotto da tastiera.

L'istruzione CMD 4,11 registra il record sul nastro magnetico.

Infine, l'ultima istruzione CMD registra un blocco lungo di controllo:

```
FILE

0010 DIM A$(10)
0020 DCL 80A$(0)
0030 BUFFER #4,80
0040 CMD #4,7,5,8
0050 FOR I=1 TO 10 STEP 1
0055 INPUT A$(I)
0060 SEND #12,A$(I)
0070 CMD #4,11
0080 NEXT I
0090 CMD #4,5,5,5,5,5
0100 END

END OF LISTING
```

Esempio 2: La routine seguente legge 10 record da un nastro magnetico di una cassetta del CTU 1010 e li stampa sul tabulato della stampante integrata e sul tabulato di una stampante esterna (PR 1220). In questo caso sono stati utilizzati due canali IPSO (canale zero per CTU 1010 e canale uno per PR 1220). L'istruzione 10 assegna 80 byte di memoria principale al canale zero; mentre l'istruzione 20 assegna 85 byte di memoria principale al canale uno.

L'istruzione 30 trasmette al CTU 1010 i comandi operativi 4,6 e 8 (rispettivamente: inizia nastro, ricerca il successivo blocco di controllo, azzerà il pointer).

L'istruzione 50 trasmette al CTU 1010 i comandi operativi 9,10 e 8 (rispettivamente: registra il valore del pointer, leggi un blocco ed azzerà il pointer).

L'istruzione 60 riceve nella variabile A\$ i dati letti dal nastro magnetico.

Infine l'istruzione 75 stampa sul tabulato della stampante PR 1220 il record letto dal nastro magnetico. Si osservi la presenza, nella istruzione suddetta, di CHR\$(10) che è il carattere di controllo LF (stampa il contenuto del buffer ed esegui una interlinea), vedi la relativa scheda per la PR 1220 nell'Appendice B.

```
FILE

0010 BUFFER #4,80
0012 BUFFER #31,85
0020 DCL 80A$
0030 CND #4,4,6,8
0040 FOR I=1 TO 10 STEP 1
0050 CND #4,9,10,8
0060 RECEIVE #4,A$
0070 PRINT A$
0075 SEND #31,A$+CHR$(10)
0080 NEXT I
0090 END

END OF LISTING
```

```
STRINGA=1*****
STRINGA=2*****
STRINGA=3*****
STRINGA=4*****
STRINGA=5*****
STRINGA=6*****
STRINGA=7*****
STRINGA=8*****
STRINGA=9*****
STRINGA=10*****
```

Tabulato della stampante integrata

```
STRINGA=1*****
STRINGA=2*****
STRINGA=3*****
STRINGA=4*****
STRINGA=5*****
STRINGA=6*****
STRINGA=7*****
STRINGA=8*****
STRINGA=9*****
STRINGA=10*****
```

Tabulato della stampante PR 1220

Esempio 3: Riportiamo nel seguito il listing e l'esecuzione di un programma più ampio che registra su di un nastro magnetico inserito nella unità CTU1010 dei blocchi di record dati. Nel listing compaiono i commenti relativi alle funzioni eseguite dalle diverse parti del programma per cui ci limitiamo ad osservare che ogni blocco è separato dal successivo con dei gap brevi e la fine registrazione è segnalata con la registrazione di un gap lungo.

Ad inizio di ogni blocco, il primo byte specifica il numero di record logico contenuto nel blocco (che è di 256 byte); il secondo byte specifica la lunghezza, in byte, del record che segue ed ogni record successivo nel blocco è preceduto da un byte che specifica la lunghezza del record.

```

FILE      CTUWR2

0010 REM ***** CTU1010 OUTPUT DRIVER - BLOCKED RECORDS*****
0020 REM
0030 BUFFER #1,256
0040 DCL 256(A$,B$)
0050 DCL SINGLE
0060 REM
0070 REM ***** CONSTANTS SET UP *****
0080 REM
0090 LET P1=1
0100 LET P0=P1-P1
0110 LET P2=P1+P1
0120 LET P3=P2+P1
0130 LET P4=P3+P1
0140 REM
0150 REM ***** REWIND *****
0160 REM
0170 CMD #P4,P4 AND GO
0180 TEST #P4
0190 IF IOC(6)=P0 THEN 320
0200 REM
0210 REM ***** CTU1010 OFF OR LOCAL - OPERATOR CALL *****
0220 REM
0230 BEEP
0240 DISP "TURN ON AND RESET CTU1010"
0250 DELAY 2
0260 DISP ""
0270 GOTO 170
0280 REM
0290 REM ***** SET UP OF T=NUMBER OF RECORDS IN A BLOCK, S=SPACE AVAILABLE *****
0300 REM ***** B$ = PHYSICAL BLOCK *****
0310 REM
0320 LET T=P0
0330 LET S=254
0340 LET B$=""
0350 REM
0360 REM ***** WAIT UNTIL REWIND IS COMPLETED *****
0370 REM
0380 CMD #12,P3
0390 REM
0400 REM ***** INPUT OF RECORD *****
0410 REM
0420 DISP "INPUT STRING":
0430 INPUT A$
0440 LET L=LEN(A$)
0450 IF L=P0 THEN 660
0460 REM
0470 REM ***** IF SPACE AVAILABLE, PUT RECORD INTO BLOCK *****
0480 REM
0490 IF L>254 THEN 780
0500 IF L>S THEN 610

```


Esempio 4: Riportiamo nel seguito un esempio di programma che legge record di dati registrati su un nastro magnetico (unità CTU 1010) in blocchi di 256 byte secondo la struttura descritta nell'esempio precedente. Il listing del programma contiene diversi commenti per cui ci limitiamo a segnalare di ricorrere alla relativa scheda riportata in appendice B per l'interpretazione del significato dei comandi ed alla Tabella 1.1 e alla pagina 3-5 per l'interpretazione del significato dei sondaggi effettuati mediante la funzione IOC. Al termine del listing è riportata la stampa dei dati letti dal nastro magnetico.

FILE

```

0010 REM ***** CTU1010 INPUT DRIVER * BLOCKED RECORDS *****
0020 REM
0030 BUFFER #1,256
0040 DCL SINGLE
0050 DIM N(1)
0060 DCL 254A$,256B$,10$
0070 REM
0080 REM ***** CONSTANTS SET UP *****
0090 REM
0100 LET P1=1
0110 LET P0=P1-P1
0120 LET P2=P1+P1
0130 LET P3=P2+P1
0140 LET P4=P3+P1
0150 REM
0160 REM ***** REWIND TO BOT *****
0170 REM
0180 CMD #4,4 AND GO
0190 TEST #4
0200 IF IOC(6)=0 THEN 320
0210 REM
0220 REM ***** CTU 1010 OFF OR LOCAL - OPERATOR CALL *****
0230 REM
0240 BEEP
0250 DISP "TURN ON AND RESET CTU 1010"
0260 DELAY 2
0270 DISP ""
0280 GOTO 180
0290 REM
0300 REM ***** WAIT END OF REWIND *****
0310 REM
0320 CMD #12,3
0330 REM
0340 REM
0350 REM ***** READ BLOCK *****
0355 REM
0360 CMD #4,10
0370 CMD #12,3 AND GO
0380 WAIT #12
0390 IF IOC(4)=1 THEN 710
0400 REM
0410 REM ***** IF EOF GOTO END OF PROGRAM *****
0420 REM
0430 RECEIVE #4,B$
0440 REM
0450 REM ***** T=NUMBER OF LOGICAL RECORDS *****
0460 REM
0470 LET C$=EXT$(B$,1,1)
0480 CONVERT C$ TO N LENGTH P1
0490 LET T=N(P1)

```




4. IMPIEGO DELLA TASTIERA INTEGRATA COME PERIFERICA ESTERNA

Come abbiamo visto nel manuale "P6060 - Manuale generale", la tastiera integrata può essere utilizzata per fornire dati al programma utente che sono richiesti da una istruzione INPUT o RKB o, quando il sistema è nello STATO DI DEBUGGING, per visualizzare il valore delle variabili di programma e per assegnare ad esse dei valori.

Tuttavia la tastiera integrata può anche essere utilizzata come una periferica esterna ed in questo caso il programma utente può riferirsi ad essa attraverso il relativo canale di I/O e le istruzioni BASIC comuni all'I/O generale descritte nel capitolo 1 che riprenderemo nei paragrafi successivi.

Oltre alle prestazioni descritte nel Manuale Generale suddetto, il programma utente può visualizzare sul display una stringa di caratteri che può essere modificata utilizzando la tastiera.

Inoltre il programma utente può richiedere dei dati da tastiera senza che ne venga sospesa l'esecuzione.

Canale di I/O

Nella figura 4-1 sono indicate le parti logiche che intervengono nell'impiego della tastiera integrata come periferica esterna, per quanto riguarda l'interazione tra tastiera, buffer da tastiera e display si veda la figura 2-5 e relativa spiegazione nel Manuale Generale.

Eseguendo una istruzione RECEIVE si abilita la tastiera ad immettere dati, attraverso il canale di I/O (vedi figura), in un buffer definito da programma mediante una istruzione BUFFER. I dati sono contemporaneamente visualizzati sul display e trasferiti dal buffer suddetto in memoria principale quando è premuto il tasto END OF LINE.

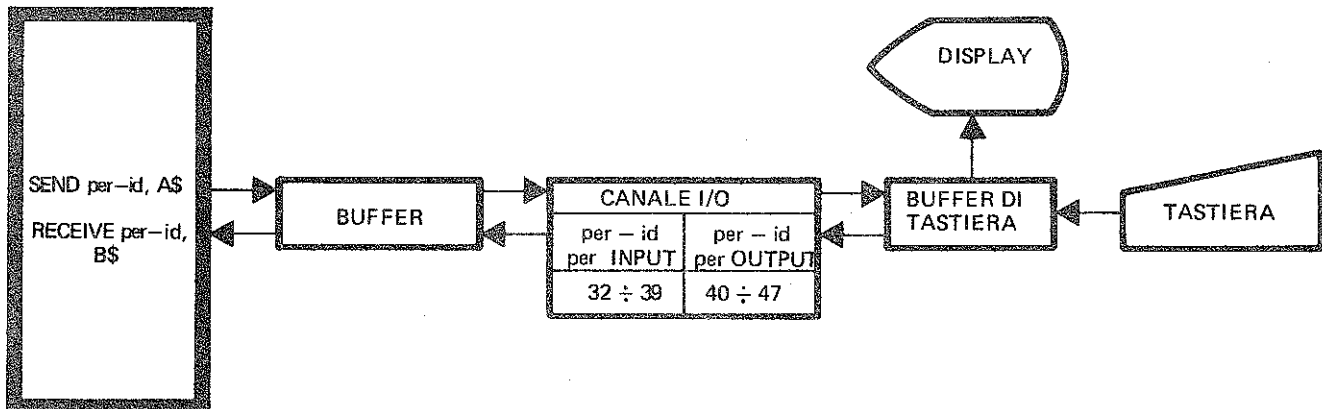


Figura 4-1 Impiego della tastiera integrata come periferica esterna

Eseguendo una istruzione SEND la stringa di caratteri in essa specificata è trasferita nel buffer definito da programma mediante l'istruzione BUFFER, da qui viene successivamente trasferita nel buffer di tastiera (vedi figura 4-1) e può essere contemporaneamente visualizzata sul display. La stringa può quindi essere modificata da tastiera e ritrasmessa dal buffer suddetto in memoria principale eseguendo una successiva istruzione RECEIVE o INPUT, MAT INPUT, RKB.

Il canale di I/O indicato in figura 4-1 ha nome logico 2 e quindi ci si può riferire ad esso indicando come valore di per-id un numero intero compreso tra 32 e 47 (istruzione BUFFER). Si osservi però che se ci si riferisce ad una operazione di input, istruzione RECEIVE od istruzione TEST e WAIT per il sondaggio dell'esito delle operazioni comandate da una istruzione RECEIVE, il valore di per-id deve essere compreso tra 32 e 39. Se ci si riferisce ad una operazione di output, istruzioni SEND, CMD od istruzioni TEST e WAIT per il sondaggio dell'esito delle operazioni comandate da istruzioni SEND e CMD, il valore di per-id deve essere compreso tra 40 e 47.

Si noti infine che i valori di per-id devono essere

omogenei per istruzioni RECEIVE, SEND e CMD da un lato ed istruzioni TEST e WAIT dall'altro. Se è stata eseguita una istruzione RECEIVE con per-id = 32, le eventuali istruzioni TEST o WAIT ad essa relative dovranno avere per-id = 32. Così se è stata eseguita una istruzione SEND con per-id = 40, le eventuali istruzioni TEST o WAIT ad essa relative dovranno avere per-id = 40. Infine se è stata eseguita una istruzione CMD con per-id = 47, le eventuali istruzioni TEST o WAIT ad essa relative dovranno avere per-id = 47.

Istruzioni BASIC

Le istruzioni BASIC che permettono l'impiego della tastiera integrata come periferica esterna sono:

```
BUFFER # per-id, buffer-size
CMD # per-id, command-code[,command-code]...[AND GO]
INTERRUPT ENABLE(E,funam,intmsk[,priort])
RECEIVE # per-id, string-var[AND GO]
SEND # per-id, string-exp[AND GO]
TEST # per-id
WAIT # per-id
e la funzione di sistema IOC (num-exp).
```

Nei paragrafi che seguono verrà descritta la funzione ad esse relativa.

Istruzione BUFFER

E' una istruzione dichiarativa che alloca in memoria principale un registro di transito, buffer, che può ricevere dati dalla tastiera od il cui contenuto può essere modificato da tastiera.

La capacità del buffer è specificata, in byte, con l'operando buffer-size. Poichè dalla tastiera non si possono introdurre più di 80 caratteri, è bene non specificare come valore di buffer-size un numero intero maggiore di 80.

Come si è già visto per-id può essere un numero compreso tra 32 e 47.

Istruzione CMD

La funzione di questa istruzione da un punto di vista generale è descritta a pagina 2-5. Per il significato di AND GO si veda la suddetta descrizione; nel seguito diamo una descrizione del significato dei codici che possono essere specificati come operando command-code.

Codice	Funzione
Ø	Non opera
1	Permette la visualizzazione sul display del contenuto del buffer
2-15	Non sono utilizzati
16	Disabilita la ricezione di caratteri dalla tastiera
17-31	Non sono utilizzati

Tabella 4-1 Codici di comando per l'impiego della tastiera integrata come periferica esterna

Come si è già visto per-id può essere un numero compreso tra 40 e 47.

Istruzione INTERRUPT ENABLE

Questa istruzione è descritta nel capitolo 2; qui ricordiamo che il bit che indica, nella maschera intmsk, che è richiesta una interruzione da parte della tastiera, è il terzo bit da sinistra. La causa che dà luogo ad un'interruzione è la digitazione del solo tasto di END OF LINE in risposta ad un'istruzione RECEIVE.

Istruzione RECEIVE

La descrizione di questa istruzione da un punto di vista generale è riportata nella pagina 2-7; vogliamo qui riportare alcune note sull'impiego di essa nel caso del canale di I/O riservato alla tastiera.

Quando è eseguita una istruzione RECEIVE senza l'opzione AND GO, l'esecuzione del programma utente è sospesa ed il sistema attende che l'utente introduca una stringa di caratteri dalla tastiera. La luce di console ON-LINE si accende. I caratteri sono introdotti dalla tastiera secondo le regole già descritte nel Manuale Generale. Non appena l'utente preme il tasto END OF LINE, i caratteri registrati nel buffer di tastiera sono assegnati alla variabile stringa specificata nella istruzione RECEIVE, il buffer è svuotato, e l'esecu-

zione del programma riprende dalla successiva istruzione. La luce di console ON LINE si spegne.

Quando è eseguita una istruzione RECEIVE contenente l'opzione AND GO, la tastiera è abilitata per l'introduzione di una stringa di caratteri. La luce di console ON LINE si accende per avvertire l'operatore che può introdurre dei caratteri da tastiera. L'esecuzione del programma non è sospesa.

I caratteri sono introdotti dalla tastiera secondo le regole descritte nel Manuale Generale. Quando è premuto il tasto **END OF LINE** la luce di console ON LINE si spegne, i caratteri contenuti nel buffer non sono assegnati alla stringa di caratteri specificata nella istruzione RECEIVE ma l'assegnazione avverrà quando sarà eseguita una successiva istruzione riferita al medesimo canale (quindi con per-id compreso tra 32 e 47).

Il valore che può essere specificato per per-id in una istruzione RECEIVE è compreso tra 32 e 39.

Non si possono introdurre da tastiera più caratteri di quanti sono stati allocati per il buffer con l'istruzione BUFFER. Se si introducono più caratteri di quanti sono definiti dalla lunghezza di allocazione della variabile stringa specificata nella istruzione RECEIVE, nel buffer viene registrata la stringa nulla.

Si noti che se prima che venga completamente eseguita una istruzione RECEIVE con l'opzione AND GO, il sistema commuta nello STATO DI DEBUGGING (esecuzione di una istruzione STOP) oppure è eseguita una istruzione INPUT, MAT INPUT od RKB, la luce di console ON LINE rimane accesa mentre la luce di console RUNNING diventa fissa e la tastiera è abilitata a soddisfare la richiesta conforme a tali situazioni: assegnazione di valori alle variabili di programma o richiesta del valore delle variabili di programma, nel caso di sistema nello STATO DI DEBUGGING; assegnazione di un valore ad una o più variabili di programma specificate nella istruzione INPUT; assegnazione dei valori alle componenti di una variabile multipla specificata nella istruzione MAT INPUT; assegnazione di un valore alla variabile specificata nella istruzione RKB. Non appena il sistema esce dallo STATO DI DEBUGGING o le suddette istruzioni sono eseguite, tasto **END OF LINE** premuto,

la luce di console RUNNING ritorna a pulsare e l'utente deve reintrodurre i caratteri che aveva eventualmente digitato in risposta alla istruzione RECEIVE suddetta.

Istruzione SEND

Quando è eseguita l'istruzione SEND la stringa di caratteri corrispondente al valore di string-exp è registrata nel buffer definito con l'istruzione BUFFER, e da qui è trasferita nel buffer di tastiera sostituendo una eventuale stringa in esso già presente.

L'esecuzione della istruzione SEND non provoca la visualizzazione sul display della stringa di caratteri in esso specificata; per ottenerne la visualizzazione da programma si deve eseguire successivamente l'istruzione CMD con il codice di comando 1. Il pointer appare sul display alla fine della stringa di caratteri ed il contenuto del buffer può essere modificato utilizzando i tasti di editing della tastiera; dalla tastiera si possono anche introdurre nel buffer altri caratteri purchè il numero totale di essi non sia superiore ad 80.

Per assegnare ad una variabile di programma il contenuto del buffer si deve eseguire una istruzione RECEIVE o INPUT, MAT INPUT, RKB.

Se la stringa di caratteri corrispondente al valore di string-exp ha più di 80 caratteri, vengono trasferiti nel buffer i primi 80 caratteri.

Si noti che il valore che può essere specificato per l'operando per-id è compreso tra 40 e 47.

Se si esegue una istruzione SEND ed il valore corrispondente a string-exp è la stringa nulla il contenuto del buffer di tastiera non è rimosso mentre lo stesso viene svuotato se si esegue una istruzione SEND con un valore di string-exp costituito da uno spazio (blank).

Istruzione TEST

Trasferisce nel Deposito di Stato Corrente le informazioni che definiscono la situazione del canale di I/O (vedi pagina 4-7) rilevabili da programma utente mediante la funzione di sistema IOC (num-exp).

Istruzione WAIT

Trasferisce nel Deposito di Stato Corrente le informazioni che definiscono la situazione del canale di I/O (vedi pagina 4-7) dopo che la relativa istruzione di I/O è stata eseguita. Dette informazioni sono rilevabili da programma utente mediante la funzione di sistema IOC (num-exp).

Funzione di sistema
IOC(num-exp)

Il valore restituito da IOC(6), IOC(4), IOC(3), IOC(2) e IOC(1) è sempre zero mentre il valore restituito da IOC(5) è sempre uno.

Il valore restituito da IOC(num-exp) quando il valore di num-exp, arrotondato all'intero più prossimo, è uguale a zero, minore di zero o maggiore di 8, non è significativo.

Se il valore ritornato da IOC(7) o IOC(8) è uno, non si è ancora risposto ad un'istruzione RECEIVE con l'opzione AND GO.

Salto della tastiera

Come abbiamo visto durante l'esecuzione di un programma BASIC, la tastiera può essere abilitata da diverse istruzioni: INPUT, MAT INPUT, RKB, RECEIVE o SEND, oppure per effetto della commutazione del sistema nello STATO DI DEBUGGING. L'utente può capire a quale di queste richieste il suo input risponde osservando le luci di console ed il display e consultando la seguente tabella di stato.

LUCE RUNNING	LUCE ON LINE	LUCE CONT	LA RICHIESTA DI INPUT PROVIENE DA
FISSA	SPENTA	ACCESA	Istruzione INPUT, MAT INPUT o RKB
FISSA	SPENTA	SPENTA	STATO DI DEBUGGING
LAMPEGGIA	ACCESA	ACCESA	Istruzione RECEIVE con opzione AND GO
FISSA	ACCESA	ACCESA	Istruzione INPUT, MAT INPUT o RKB ed istruzione RECEIVE con opzione AND GO
FISSA	ACCESA	SPENTA	STATO DI DEBUGGING ed istruzione RECEIVE con opzione AND GO

Tabella 4-2 Stato della tastiera

Esempi di programmazione

Esempio 1: Il programma seguente registra sul file ad accesso diretto, ARCHIV, un insieme di record utilizzando per l'input da tastiera l'istruzione RECEIVE.

```
.  
LIST  
FILE      CRE  
  
0010 DCL 80A$  
0020 BUFFER #32,80  
0030 FILES ARCHIV  
0040 SETM :1 TO 1  
0050 PAD A$,32  
0060 WRITE :1,A$ EOF 80  
0070 GOTO 60  
0080 SETM :1 TO 1  
0090 DISP "INTRODUCI RECORD"  
0100 RECEIVE #32,A$  
0110 PRINT A$  
0120 PAD A$,32  
0130 WRITE :1,A$  
0140 GOTO 90  
0150 END
```

END OF LISTING

```
RUN  
**** FORMALLY CORRECT PROGRAM ****  
INTRODUCI RECORD  
CHIAVEGATO ITALO VIVO  
INTRODUCI RECORD  
FISANI FEDERICO VIVO  
INTRODUCI RECORD  
VIGIAK RODOLFO VIVO  
INTRODUCI RECORD
```

Esempio 2: Il seguente programma legge i record registrati con il precedente programma nel file ad accesso diretto ARCHIV. L'istruzione 150 permette all'utente di decidere se aggiornare o meno il record visualizzato. Se in risposta all'istruzione si preme solo **END OF LINE** il record non viene aggiornato e viene letto e visualizzato il record successivo. Se si introduce E e si preme **END OF LINE** il record viene visualizzato di nuovo per essere modificato (istruzioni 200 e 220). Il record modificato, istruzione 240, è stampato (istruzione 250) prima di essere registrato nella precedente posizione nel file (istruzione 290).

```

FILE      KBCH

0010 REM ALLOCA BUFFER E STRINGHE
0020 DCL 80(A$,A1$)
0030 BUFFER #32,80
0040 REM APRI FILE DA AGGIORNARE
0050 FILES ARCHIV
0060 REM SCANSIONE DEL FILE
0070 FOR I=1 TO 10000 STEP 21
0080 REM LETTURA DEL FILE
0090 SETW :1 TO I
0100 READ :1,A$ EOF 310
0110 REM VISUALIZZA RECORD IN MODO NON EDITABILE
0120 DEPAD A$,32
0130 DISP A$
0140 REM RICHIESTA DI MODO
0150 RECEIVE #32,B$
0160 IF B$<>"E" THEN 300
0170 REM MODO EDITING
0180 REM
0190 REM RENDI EDITABILE IL RECORD
0200 SEND #40,A$
0210 REM VISUALIZZA RECORD EDITABILE
0220 CMD #40,1
0230 REM RICEVI RECORD EDITATO
0240 RECEIVE #32,A1$
0250 PRINT A1$
0260 PAD A1$,32
0270 REM AGGIORNA IL CONTENUTO DEL FILE
0280 SETW :1 TO I
0290 WRITE :1,A1$
0300 NEXT I
0310 END

```

END OF LISTING

```

RUN
CHIAVEGATO ITALO VIVO
PISANI FEDERICO VIVO
PISANI FEDERICO NORTO
VICIAC RODOLFO VIVO

```

