GDOS

80

GRIFO® Disk Operating System

80 family

MANUALE D'USO

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
Email: grifo@pt.tizeta.it
FAX +39 (0)51 89 36 61
Tel. +39 (0)51 89 20 52 (4 lin.r.a.)

GDOS 80 Edizione 3.2 *Rel. 15 Luglio 1996*

GDOS

80

GRIFO® Disk Operating System
80 family

MANUALE D'USO

Il **GDOS** è un completo pacchetto software, realizzato dalla **GRIFO®**, per il carteggio industriale **ABACO®**.
Il suffisso **80** indica la versione per le schede aventi CPU con codice Z80 compatibile.
Questo manuale costituisce una completa guida utente per il corretto uso del **GDOS** (**GRIFO® D**isk **O**perating **S**ystem).
Grazie al **GDOS,** è possibile utilizzare per lo sviluppo dei vari applicativi, una notevole varietà di linguaggi evoluti e risorse di debugger, non disponibili facilmente con altri pacchetti software.

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
Email: grifo@pt.tizeta.it
FAX +39 (0)51 89 36 61
Tel. +39 (0)51 89 20 52 (4 lin.r.a.)

GDOS 80 Edizione 3.2 *Rel. 15 Luglio 1996*

Vincoli sulla documentazione grifo® Tutti i Diritti Riservati

IMPORTANTE

Tutte le informazioni contenute in questo manuale sono state accuratamente verificate, ciononostante **grifo**® non si assume nessuna responsabilità per danni diretti o indiretti a cose e/o persone derivanti da errori tecnici ed omissioni o dall'uso del presente manuale, del software o dell'hardware ad esso associato.
grifo® altresi si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l'intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo **grifo**®.
Per le informazioni specifiche sui componenti montati sulla scheda, l'utente deve fare riferimento ai Data Book delle case costruttrici o delle seconde sorgenti.

Marchi Registrati

INDICE GENERALE

INDICE DELLE FIGURE

INTRODUZIONE

Questo manuale fornisce tutte le informazioni hardware e software per consentire all'utente il miglior utilizzo del pacchetto software **GDOS** (**GRIFO® D**isk **O**perating **S**ystem). Al fine di non incontrare problemi nell'uso di questo software, è conveniente che l'utente legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase per rintracciare più facilmente le informazioni necessarie si può fare riferimento al sommario ed all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

GENERALITA'

Il **GDOS** è un pacchetto software realizzato per tutte le schede basate su un microprocessore della vasta famiglia **Z80**, appartenenti al carteggio industriale della **GRIFO®**.
Tale pacchetto consente di operare con queste schede sfruttando un ambiente di lavoro molto evoluto, che non necessita nemmeno di una conoscenza approfondita di quello che è l'hardware che si utilizza. Con il **GDOS** l'utente può sfruttare linguaggi evoluti per la programmazione ed avere a disposizione molte risorse normalmente non disponibili con altri pacchetti software.
E' possibile realizzare, testare, ed installare ogni tipo di programma che l'utente realizza, ottenendo rapidamente ed in modo molto comodo ed efficace delle specifiche applicazioni.
Il **GDOS** è un semplice pacchetto software che può essere paragonato ad un elementare, ma efficace, sistema operativo per il settore industriale. Esso trae origine dal CP/M della Digital Research, con cui mantiene la compatibilità, le modalità di programmazione ed utilizzo. Come tutti i sistemi operativi, anche il **GDOS** si occupa di effettuare un interfacciamento ad alto livello tra l'utente e l'hardware da utilizzare. Con questo pacchetto si ha quindi la possibilità di sfruttare tutte le risorse hardware del sistema prescelto, direttamente con istruzioni, comandi, linguaggi di programmazione ad alto livello, senza doversi preoccupare di sviluppare firmware di gestione specifico. La notevole potenza e duttilità del **GDOS** é quella di poter usufruire di una ricca serie di ambienti di programmazione (teoricamente tutti i programmi e/o linguaggi in grado di operare in CP/M), e di avere la possibilità di sfruttare al meglio le risorse di un calcolatore esterno, che semplifica notevolmente la fase di sviluppo e messa a punto dell'applicazione.
Ad esempio il **GDOS** si occupa della gestione di risorse hardware come le linee seriali, le stampanti, le memorie di massa, le interfaccie operatore, ecc rendendole disponibili all'utente tramite le apposite istruzioni del linguaggio di programmazione scelto. Normalmente il sistema operativo fà uso di una piccola quantità di memoria e condivide l'uso di una linea seriale; qualora queste caratteristiche diventino limitative, esistono delle soluzioni hardware che risolvono il problema (per maggiori informazioni contattare la **Grifo®**).
Il **GDOS** é un pacchetto software composto da vari sottogruppi indipendenti e non, che insieme costituiscono un comodo ambiente di sviluppo, provvisto di: commenti, guide, aiuti in linea, esempi, ecc. Ognuno di questi sottogruppi é dotato di un proprio numero di versione a cui si deve sempre fare riferimento in caso di richiesta di informazioni alla **GRIFO®**; in questo manuale invece é riportata una descrizione completa di tutto il pacchetto seguendo una suddivisione logica relativa al significato dei singoli componenti.
Vista la naturale evoluzione dei pacchetti software, si faccia sempre attenzione all'eventuale presenza del file "READ.ME" nel disco di lavoro. Tale file coincide con la serie di aggiunte, modifiche e miglioramenti apportati nel tempo, a tutto il pacchetto software e non ancora riportati sul manuale; se é presente lo si deve esaminare, stampare ed allegare al presente manuale.

MATERIALE NECESSARIO

Viene di seguito riportata una breve descizione del materiale (hardware e software), necessario per operare con il **GDOS**:

SCHEDA DI CONTROLLO BASATA SULLO Z80

Coincide con la scheda di controllo appartenente al carteggio industriale **GRIFO®**, basata sui microprocessori della famiglia Z80 come: **GPC® 80F; GPC® 81F; GPC® 011; GPC® 15A; GPC® 15R; GPC® 15T**; **GPC® 153; GPC® 183;** ecc.
N.B. In questo manuale si utilizza l'indicazione: **scheda remota** per far riferimento ad una delle strutture hardware della **GRIFO®** sopra riportate.
La scheda remota, indipendentemente dalle richieste dell'applicazione da realizzare, deve essere dotata di:

EPROM o FLASH EPROM con sistema operativo **GDOS**
almeno 64 KByte di RAM
una linea seriale asincrona in RS 232

Quanto sopra riportato è da intendersi come struttura minima di lavoro, infatti lo stesso sistema può essere espanso aumentando quindi le sue potenzialità. Il **GDOS** infatti, gestisce sempre con un'interfaccia ad alto livello, le seguenti sezioni della scheda remota:

1) 1 Linea seriale di consolle = indispensabile in tutti i casi in cui si utilizza il **GDOS** anche per lo sviluppo ed il debug dell'applicazione.
2) 1 Linea seriale ausiliaria = su cui viene gestita ad alto livello la ricezione e la trasmissione di caratteri secondo un protocollo preimpostato.
3) Possibilità di utilizzare una comunicazione parallela tra P.C. e scheda remota, in modo da accellerare le fasi di debug e sviluppo.
4) 14 Linee di I/O TTL = gestite ad alto livello per il comando di una stampante parallela (interfacciata ad esempio con la **IAC 01** o la **DEB 01**).
5) EPROM o FLASH EPROM eccedente i 16 KByte = gestita ad alto livello come ROM DISK, in cui i dati sono prelevati sotto forma di files.
6) RAM eccedente i 64 KByte = gestita ad alto livello come RAM DISK, in cui i dati sono salvati e prelevati sotto forma di files.
7) EEPROM seriale di bordo gestita ad alto livello come un buffer in cui i dati sono salvati e prelevati sotto forma di bytes.
8) 16 Linee di I/O TTL = gestite ad alto livello per la gestione di una RAM CARD PCMCIA (interfacciata con la **MCI 64**).
9) 16 Linee di I/O TTL = gestite ad alto livello per la gestione di una interfaccia operatore locale basata su display alfanumerici, LEDs e tastiera a matrice (ad esempio **QTP 24P**, **KDL x24**, **KDF 224**, ecc).

La scelta della configurazione della scheda di controllo deve quindi avvenire in relazione alle specifiche esigenze dell'applicazione che deve essere sviluppata.

EPROM O FLASH EPROM GDOS

Il codice del sistema operativo viene sempre fornito in EPROM (**GDOS**) o FLASH EPROM (**FGDOS**) pronta ad essere montata sulla scheda remota scelta. In caso di acquisto contemporaneo di scheda remota e **GDOS**, il dispositivo di memoria viene fornito già montato sulla scheda. Sull'etichetta del dispositivo sono presenti tutte le informazioni relative al tipo di scheda remota ed alla versione del sistema operativo e l'eventuale contenuto della ROM DISK a sua volta presente nello stesso dispositivo.

PERSONAL COMPUTER

Un Personal Computer IBM (o compatibile) provvisto di un floppy disk, almeno 640K RAM e di una linea seriale in RS 232 secondo normative V24, provvisto di sistema operativo MS-DOS versione 3.3 e sucessive. La presenza di un hard disk é consigliata, anche se non indispensabile, al fine di velocizzare tutte le operazioni di accesso ai files.
Questo sistema non è indispensabile ma viene normalmente utilizzato, in quanto rende possibile e facilita motissimo, lo sviluppo ed il debug del programma di gestione che l'utente deve realizzare. In particolare il P.C. collegato in seriale alla linea di consolle della scheda remota, funziona da terminale con cui l'utente può interagire con il **GDOS** in esecuzione sulla scheda remota ed allo stesso tempo mette a disposizione di quest'ultima le sue risorse (memorie di massa e stampante) in modo trasparente.

SOFTWARE DI LAVORO

Utilizzando il Personal Computer per lo sviluppo e la messa a punto del programma di gestione, è necessario disporre di una serie di programmi e files, presenti nel disco di distribuzione del **GDOS**. Tra questi si ricordano i programmi di emulazione terminale intelligente (**GET80.EXE**), il programma di generazione dell'immagine della ROM DISK (**FGROM.G80**), tutti i programmi di utility del **GDOS** (DIR.G80, ERA.G80, REN.G80, ecc.) e lo stesso sistema operativo **GDOS**.
Per maggiori informazioni a riguardo di questo software si veda il capitolo "UTILIZZO DEL **GDOS**".

SOFTWARE DI PROGRAMMAZIONE

Coincide con le strutture di lavoro che l'utente deve utilizzare per la scrittura e la prova del programma di gestione da sviluppare. In particolare con software di programmazione s'intendono tutti i linguaggi di programmazione (BASIC interpretati e compilati; PASCAL; FORTRAN; C; ASSEMBLY; MODULA 2; ecc) i programmi di prova e verifica (MONITOR-DEBUGGER) e gli eventuali editor, assemblatori e linker per ottenere la versione eseguibile del programma da sviluppare.
Per tutte le informazioni che riguardano questo software (che deve essere specificatamente ordinato), si faccia riferimento agli appositi manuali ricevuti assieme al pacchetto **GDOS**.

CAVO DI COMUNICAZIONE SERIALE

Nel caso di utilizzo della linea seriale di consolle della scheda remota per lo sviluppo ed il debug del software di gestione, si deve provvedere a collegare il dispositivo di consolle in modo opportuno. Ricordando che la linea di consolle della scheda remota coincide sempre con la linea seriale A della scheda utilizzata e che il **GDOS** gestisce oltre ai due segnali di ricezione e trasmissione (TxD e RxD) anche i due handshake di abilitazione (/CTS e /RTS), il collegamento deve avvenire seguendo le normative V24 del C.C.I.T.T.
Nel caso in cui il dispositivo di consolle coincida con il Personal Computer, il collegamento da effettuare deve essere quello rovesciato (DTE <->DCE), di seguito descritto:

P.C. Femmina DB25	DB9		CN? scatolino scheda remota	
TxD = 2	3	>——————>	9	= RxD
RxD = 3	2	<——————<	10	= TxD
DSR = 6	6	<——————<	8	= RTS
DTR = 20	4	>——————>	7	= CTS
GND = 7	5	<——————>	2	= GND

FIGURA 1: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE A SCATOLINO 16 VIE

P.C. Femmina DB25	DB9		CN? vaschetta 9 vie scheda remota	
TxD = 2	3	>——————>	2	= RxD
RxD = 3	2	<——————<	3	= TxD
DSR = 6	6	<——————<	7	= RTS
DTR = 20	4	>——————>	8	= CTS
GND = 7	5	<——————>	5	= GND

FIGURA 2: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE A VASCHETTA 9 VIE

P.C. Femmina DB25	DB9		CN? vaschetta 25 vie scheda remota	
TxD = 2	3	>——————>	3	= RxD
RxD = 3	2	<——————<	2	= TxD
DSR = 6	6	<——————<	4	= RTS
DTR = 20	4	>——————>	5	= CTS
GND = 7	5	<——————>	7	= GND

FIGURA 3: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE A VASCHETTA 25 VIE

P.C. Femmina DB25	DB9		CN? plug 6 vie scheda remota	
TxD = 2	3	>——————>	5	= RxD
RxD = 3	2	<——————<	2	= TxD
DSR = 6	6	<——————<	3	= RTS
DTR = 20	4	>——————>	4	= CTS
GND = 7	5	<——————>	6	= GND

FIGURA 4: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE PLUG 6 VIE

Con le indicazioni **CN? scatolino scheda remota** e **CN? vaschetta scheda remota** s'intendono i connettori standardizzati per la comunicazione seriale, della **GRIFO®**. Per questo è stata utilizzata una indicazione generica; l'utente potrà completare l'indicazione facendo riferimento al manuale tecnico della scheda remota che stà utilizzando. Al fine di velocizzare la fase di connessione eliminando allo stesso tempo la necessità di dover realizzare un cavo di comunicazione, la **GRIFO®** é in grado di offrire i cavi di comunicazione seriale già pronti, per qualsiasi tipo di scheda e P.C.

DISCHI DISTRIBUITI

Il pacchetto software **GDOS** è normalmente composto da una EPROM o FLASH EPROM montata sulla scheda remota basata sullo Z80 e da una serie di dischi che contengono tutta la struttura software necessaria per il lavoro.
Per quanto riguarda i dischi relativi ai linguaggi di programmazione (ZBASIC, PASCAL, MODULA 2, ASSEMBLY, ecc.) si ricorda che questi non sono di libera distribuzione e che possono essere accompagnati dal manuale e dal/dai dischi originali distribuiti dalla ditta che li ha prodotti: quest'ultimi dischi non devono essere utilizzati con il **GDOS.** Essi sono sostituiti da quelli rilasciati dalla **GRIFO®**.
Di seguito viene riportata una breve descrizione di cosa è presente nei dischi ricevuti; si ricorda che tutti i files con l'estensione **.G80**, sono programmi eseguibili dalla scheda tramite il **GDOS**.

DISCO DI LAVORO

Contiene tutto il software di lavoro e la serie di software di programmazione. Il contenuto del dischetto è organizzato in varie directory, in modo da semplificare e rendere più chiara tutta la struttura; in fase di utilizzo del GDOS, come consigliato nel capitolo "COME INIZIARE", è conveniente copiare tutti i files in una sola directory di lavoro. Qui di seguito viene riportata una semplice lista delle directory e dei files presenti nel disco di lavoro; per avere maggiori informazioni sull'utilizzo ed il significato di questi, si faccia riferimento ai paragrafi dei capitoli seguenti:

Root:
Contiene tutti i programmi usati più frequentemente quando si opera con il **GDOS**, ovvero la struttura di lavoro principale:

GET80.EXE -> Programma di supporto al **GDOS** da eseguire sul P.C.
G80INST.EXE -> Programma di installazione per il **GET80.EXE**
DOS2GDOS.COM -> Programma per la trasformazione del formato dei files da eseguire sul P.C.
DIR.G80 -> Utility del **GDOS** per ottenere il diretorio di un disco.
ERA.G80 -> Utility del **GDOS** per cancellare un file da un disco.
REN.G80 -> Utility del **GDOS** per rinominare un file da un disco.
COPY.G80 -> Utility del **GDOS** per copiare un file da disco a disco.
FORMAT.G80 -> Utility del **GDOS** per formattare un drive RAM DISK della scheda remota.
READ.ME -> File di documentazione con le aggiunte al presente manuale.

Directory Z80_EMUL:
Contiene un comodo e potente programma di emulazione del sistema operativo CP/M e quindi del microprocessore Z80, in ambiente MS/DOS:

Z80MU.EXE -> Programma di simulazione sistema operativo CP/M su P.C. Con tale programma possono essere eseguiti tutti i command file del **GDOS**, ma a differenza di quest'ultimo l'estensione deve essere .COM e non .G80. Per fare girare sotto Z80MU dei programmi con estensione .G80 è sufficiente rinominarli. E' da utilizzare per familiarizzare e provare i linguaggi di programmazione disponibili. E' autodocumentato ed, a richiesta, fornisce a video l'elenco dei comandi accettati e la sintassi.

Directory ROM_DISK:
Contiene la serie di programmi e files che devono essere utilizzati per salvare in EPROM o FLASH EPROM il programma applicativo realizzato dall'utente; in funzione del **GDOS** acquistato, (GDOS o FGDOS), saranno presenti alcuni dei file di seguito riportati.

GDO???XX.BIN -> Immagine binaria del sistema operativo **GDOS** per la scheda ??? nella versione XX. E'da utilizzare per creare la EPROM di lavoro. Presente solo nella versione EPROM del **GDOS**.

GHEX2.COM -> Programma per la trasformazione di un file binario nell'equivalente file nel formato HEX Intel. Da utilizzare nel caso in cui il programmatore di EPROM posseduto, non utilizzi un file binario. Presente solo nella versione EPROM del **GDOS**.

FGROM.G80 -> Programma per la programmazione della FLASH EPROM montata sulla scheda remota: consente di preparare la FLASH EPROM con una completa struttura di ROM DISK e viene eseguito sulla scheda remota. Presente solo nella versione FLASH EPROM del **GDOS**.

DISCO ZBDEMO

Contiene una struttura dimostrativa per la programmazione ad alto livello con il **GDOS,** coincidente con un BASIC compilato, più una ricca serie di programmi dimostrativi di uso generale per tutte le possibili schede remote ed altre schede.

ZBDEMO.G80 -> Linguaggio dimostrativo di programmazione BASIC compilato per **GDOS**.
ZBASIC.HLP -> File per l'help in linea per il programma ZBDEMO.

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione ZBDEMO; ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.)

DISCO ZBASIC

Contiene una completa struttura di programmazione ad alto livello con il **GDOS**, coincidente con un BASIC compilato, più una ricca serie di programmi dimostrativi di uso generale per tutte le possibili schede remote ed altre schede.

ZBASIC.G80 -> Linguaggio di programmazione BASIC compilato per **GDOS**.
ZBASIC.HLP -> File per l'help in linea per il programma ZBDEMO.

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione ZBASIC; ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.)

DISCO NSB8

Contiene una completa struttura di programmazione ad alto livello con il **GDOS**, coincidente con un BASIC interpretato, più una ricca serie di programmi dimostrativi di uso generale per tutte le possibili schede remote ed altre schede.

NSB8.G80	-> Linguaggio di programmazione BASIC interpretato per **GDOS**.
BASYM.G80	-> Programma di utility per NSB8 eseguibile da **GDOS**.
BASTRA.G80	-> Programma di utility per NSB8 eseguibile da **GDOS**.
BASYM.DOC	-> Documentazione relativa al programma BASYM.G80.
BASTRA.DOC	-> Documentazione relativa al programma BASTRA.G80.

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione NSB8; ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.)

DISCO PASCAL

Contiene una completa struttura di programmazione ad alto livello con il **GDOS**, coincidente con un compilatore PASCAL, più una ricca serie di programmi dimostrativi di uso generale per tutte le possibili schede remote ed altre schede.

PASCAL.G80	-> Linguaggio di programmazione PASCAL compilato per **GDOS**
COMPILER.MSG	-> File di supporto con la messaggistica del programma PASCAL.
COMPILER.OVR	-> File di supporto per il programma PASCAL.

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione PASCAL compilato; ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.)

DISCO MODULA 2

Contiene una completa struttura di programmazione ad alto livello con il **GDOS**, coincidente con un compilatore MODULA 2, più una ricca serie di programmi dimostrativi di uso generale per tutte le possibili schede remote ed altre schede.

M2.G80	-> Linguaggio di programmazione MODULA 2 per **GDOS**.
M2.OVR	-> File di supporto per MODULA 2.
ERRMSGS.OVR	-> File di supporto con la messaggistica del programma MODULA 2.
SYSLIB.LIB	-> Libreria di base per il programma MODULA 2.
*.MCD	-> Moduli per la programmazione in MODULA 2.

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione

MODULA 2; ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.)

DISCO ASSEMBLY

Contiene una completa struttura di programmazione a basso livello con il **GDOS**, coincidente con un assemblatore,linker , debugger e disassemblatore, più una ricca serie di programmi dimostrativi di uso generale per tutte le possibili schede remote ed altre schede.

M80.G80	-> Macro-assemblatore per microprocessore Z80, quando si lavora in assembly.
L80.G80	-> Linker per microprocessore Z80, quando si sviluppa in assembly.
ZSID.G80	-> Monitor Debugger per microprocessore Z80, quando si lavora in assembly.
HIST.UTL	-> Programma di utility per ZSID.G80.
TRACE.UTL	-> Programma di utility per ZSID.G80.
REZJ1.G80	-> Disassemblatore per microprocessore Z80, eseguibile direttamente dal **GDOS**.
REZJ1.DOC	-> Documentazione relativa all'uso del REZJ1.G80.

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione ASSEMBLY; ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.)

DISCO HTC

Per ogni scheda remota é presente una directory che contiene una serie di programmi dimostrativi di uso generale che illustrano come utilizzare le risorse a bordo scheda con il linguaggio di programmazione HTC (compilatore C); ogni directory, a sua volta, può contenere una serie di sottodirectory con altri programmi dimostrativi relativi ad applicazioni ad uso generale (es. **DEB 01**, TELECONTROLLO, ecc.). In questo caso il disco contiene solo i programmi dimostrativi con i relativi file di progetto e le eventuali librerie sviluppate, ma non la struttura di programmazione che é fornita separatamente in un apposito "pacchetto software".

Nei precedenti paragrafi sono stati descritti i pacchetti di programmazione più frequentemente utilizzati, si tenga comunque conto che ne esistono altri come LISP, FORTH, C e che in caso di specifiche esigenze diverse, si può contattare direttamente la **GRIFO**®.

COME INIZIARE

In queso capitolo vengono descritte quali sono le operazioni da effettuare per un primo elementare utilizzo del pacchetto software **GDOS**. In particolare viene riportata la giusta sequenza di operazioni nel caso di utilizzo del Personal Computer come sistema di sviluppo. In questo capitolo si fa riferimento alle informazioni riportate nel precedente capitolo “MATERIALE NECESSARIO”.

1) Leggere tutta la documentazione ricevuta.

2) Predisporre la scheda remota per operare (alimentazione, verifica di configurazione, ecc.), montando la EPROM o FLASH EPROM del GDOS se non già presente.

3) Effettuare il collegamento seriale seguendo le indicazioni del paragrafo “Cavo di comunicazione”

4) Accendere il Personal Computer.

5) Creare una directory sull’hard disk del P.C. Se il P.C. è privo di hard disk, provvedere ad effettuare una copia del disco di lavoro ricevuto e passare al punto **8**.

6) Copiare nella directory creata tutto il software di lavoro ed il software di programmazione che s’intende usare e gli eventuali esempi di interesse (si veda capitolo “DISCHI DISTRIBUITI”).

7) Portarsi nella directory di lavoro creata.

8) Installare il programma **GET80.EXE**, ovvero lanciare il programma di installazione **G80INST.EXE** e fornire tutte le informazioni richieste dalla finestra che compare: il numero di linea seriale (COM) del P.C. usata per il collegamento con la scheda remota, il baud rate utilizzato per la comunicazione con la scheda remota, il tipo di monitor utilizzato (colori o bianco e nero). Per maggiori informazioni si veda il paragrafo "INSTALLAZIONE".

9) Eseguire il programma di editor ed emulazione terminale intelligente “**GET80.EXE**”, (vedere il paragrafo "GET80: PROGRAMMA DI EDITOR ED EMULAZIONE TERMINALE PER P.C.") ed attendere la comparsa della finestra di presentazione.

10) Chiudere la finestra di presentazione premendo ENTER ed infine selezionare l'opzione "Terminal" dal menù Options, con cui viene rappresentata una videata pulita con il cursore in alto a sinistra e con la seguente linea di stato sull’ultima riga del monitor:

F1 Help | TERMINAL EMULATION for GDOS - Rel. X.X - GRIFO° Tel. +39-51-892052

11) Alimentare la scheda remota: in corrispondenza di questa operazione sul monitor del P.C. deve comparire un’indicazione relativa al **GDOS** eseguito dalla scheda remota, seguita dal prompt del sistema operativo:

GDOS - Ver. X.X - Rel. XXX XXX - by GRIFO° 051 892052 Italy

N:ABACO°>

12) Lavorare con il **GDOS** secondo le modalità illustrate nel capitolo sucessivo “UTILIZZO DEL

GDOS". In questa fase ad esempio possono essere forniti i comandi diretti, lanciati programma di utility esterni, lanciati linguaggi di programmazione, ecc.
Ad esempio possono essere forniti i seguenti comandi:

N:ABACO°>**VER**<cr>
con cui viene ripresentata la linea di stato del sistema operativo del **GDOS**.

N:ABACO°>**DIR C:**<cr>
con cui viene presentato l'elenco di files presenti sulla directory attiva dell'hard disk del P.C., caricando il programma di utility DIR.G80 dal drive attuale = N = ROM DISK.

N:ABACO°>**N:DIR N:**<cr>
con cui viene presentato l'elenco di files presenti sulla ROM DISK della scheda remota, caricando il programma di utility DIR.G80 dal drive N di ROM DISK.

N:ABACO°>**N:ZBASIC**<cr>
con cui viene caricato il linguaggio di programmazione ZBASIC.G80 dal drive N di ROM DISK ammesso che sia presente su questo drive.

Per ritornare al sistema operativo MS-DOS, premere il tasto funzione F10 ed in seguito i tasti "Alt+X".

In questo capitolo, in corrispondenza delle visualizzazioni effettuate sul monitor del P.C., sono state usate delle indicazioni generichecomposte da **X**, in questi casi ognuno di questi simboli coincidecon una cifra o lettera generica, utilizzata per i numeri di versione o tipo di release.

UTILIZZO DEL GDOS

Viene di seguito riportata una descrizione del modo di utilizzo della struttura software **GDOS**. In particolare vengono riportate tutte le indicazioni che riguardano l'uso del programma di Editor e comunicazione intelligente per P.C., l'uso dei comandi del sistema operativo remoto, l'utilizzo della RAM/ROM DISK della scheda remota ed infine la creazione della EPROM o della FLASH EPROM da montare sulla scheda remota.

GET80: PROGRAMMA DI SUPPORTO PER P.C.

Il **GET80** (**Grifo® E**ditor **T**erminal **80** family) si utilizza in congiunzione al **GDOS** in esecuzione sulla scheda remota e fornisce all'utente la possibilità di editare un programma, trasferirlo da P.C. a scheda remota e viceversa, eseguirlo e quindi provarlo. Esso si occupa di due aspetti principali: la simulazione di un terminale intelligente, ovvero provvede a gestire tutte le funzioni di console come un semplice terminale (ma in più offre la possibilità di vedere e quindi gestire le risorse di memoria di massa del P.C.) e l'editor di programmi applicativi sviluppati dall'utente. Inoltre lo stesso programma mette a disposizione due utility; la prima permette di salvare una serie di stringhe definite dall'utente da utilizzare durante la fase di sviluppo del programma, per minimizzare il numero di battiture sulla tastiera del P.C. e quindi velocizzare questa fase. La seconda utility del **GET 80** invece, permette di creare l'immagine binaria del file da memorizzare in EPROM, con il programma applicativo sviluppato dall'utente o più in generale di preparare l'immagine binaria di una EPROM con ROM DISK definita dall'utente. Il tutto in modo molto facile ed immediato, senza dover essere a conoscenza dei vari indirizzi fisici della EPROM a cui scrivere i vari files.
Il programma è totalmente basato su chiamate alle funzioni dell'MS_DOS, di conseguenza può funzionare su qualsiasi sistema che utilizzi questo sistema operativo a partire dalla versione 3.3. Il **GET80** é un programma di facile utilizzo che sfrutta un'interfaccia utente ad alto livello provvista di aiuti, menù a tendina, colori, finestre di scelta a scorrimento, richiamo delle opzioni tramite tasti appositi, gestione del mouse, ecc.
Vengono di seguito riportate le caratteristiche fondamentali del programma **GET80.EXE** descritto, per quanto riguarda l'utilizzo delle risorse P.C:

Monitor:	Colori o Bianco e nero
Stampante:	Parallela
Memorie di massa:	Drive da A a C in qualsiasiasi formato gestito da MS-DOS.
Terminale:	Emulazione terminale tipo ADDS Viewpoint monocromatico.
Seriale:	COM 1, 2, 3, 4 secondo specifiche V24.
Mouse:	Microsoft compatibile con relativo driver installato.

Da notare che il P.C. che esegue il **GET80**, rimane un valido strumento di supporto solo durante la fase di realizzazione e di debug dei programmi, infatti la scheda remota può comunque operare senza il Personal, oppure tramite un tradizionale terminale con interfaccia seriale. Da questo punto di vista l'eventuale terminale da collegare alla scheda remota, deve seguire le specifiche tecniche riportate in questo manuale.

La sintassi da utilizzare per lanciare il programma è la seguente:

GET80<CR>

da fornire direttamente al sistema operativo MS - DOS.
Uuna volta lanciato, il **GET80** provvede a inizializzare il personal e si presenta con una finestra informativa al centro del monitor in cui sono riportate tutte le informazioni di versione e le generalità della **Grifo®** e quelle dell'utente specificate in fase d'installazione. Premendo il tasto ENTER o facendo click con il mouse su OK, tale finestra informativa scompare e diventa accessibile la finestra principale del programma; in questa sono disponibili cinque menù a tendina provviste di una serie di opzioni che vengono descritte nei seguenti paragrafi.

INSTALLAZIONE

Prima di utilizzare il **GET80.EXE** si deve provvedere alla sua installazione. Per questo è stata realizzato un apposito programma chiamato **G80INST.EXE** da eseguire sul P.C., che deve essere lanciato dalla stessa directory in cui si trova il **GET80.EXE**. Una volta eseguito, il **G80INST** presenta una schermata in cui vengono richiesti i cinque parametri d'installazione: il numero di seriale (COM) del P.C., il baud rate utilizzato, il tipo di monitor del P.C. (colori o bianco e nero), il nome dell'utente ed il nome della ditta utilizzatrice. Da notare che i cinque dati sopra descritti vengono richiesti solo se il **GET80.EXE** non è già stato installato e se lo é vengono richiesti solo i primi tre dati; da questo si ricava che l'installazione in termini di nome dell'utente, può essere effettuata una sola volta.
Il numero di linea seriale é un dato che deve essere scelto dall'utente in base alla configurazione del P.C. che intende usare, mentre il baud rate coincide con quello massimo che la scheda remota può gestire (vedere capitolo "SETTAGGI DEL **GDOS**").
In ogni momento si può abortire l'installazione con il tasto "Abort", oppure confermarla con il tasto "Install"; se quest'ultimo tasto viene attivato senza che il nome dell'utente e della ditta siano stati inseriti, la stessa installazione non avviene.
Se viene lanciato il **GET80.EXE** senza averlo installato, questo non viene eseguito e compare un'apposito messaggio d'errore, viceversa parte presentando anche le informazioni del nome dell'utente e della ditta.

EDITOR

Il programma **GET80.EXE** comprende un potente e versatile Editor in grado di creare file ASCII direttamente utilizzabili da **GDOS**; é provvisto di tutte le classiche funzioni di un editor con la possibilità di operare contemporaneamente anche su più file associati ad altrettante finestre di rappresentazione e di una vasta serie di opzioni che ne facilitano l'uso in tutte le condizioni. L'unico limite dell'editor é quella di non poter lavorare con finestre o files che hanno lunghezza superiore ai 64 KByte: questa dimensione è difficilmente superabile e comunque lo si risolve editando con più finestre e quindi più files. La finestra di editor coincide con la finestra principale del programma e per questo motivo, non tutti i suoi menù riguardano l'editor, bensì anche le altre funzionalità del **GET80**. Per completezza in questo paragrafo viene riportata una breve descrizione di tutti i menù e di tutte le opzioni presenti, mentre per le funzioni che non riguardano l'editor si trovano degli approfondimenti negli appositi paragrafi sucessivi.

Menù File

Opzione	*Tasto*	*Funzione*
New	-	Apre un nuovo file da editare con il nome "Untitled"
Fast Open...	F3	Apre un file in modalità veloce, ponendolo nella finestra di editor
Open ...	F4	Apre un file in modalità normale, ponendolo nella finestra di editor
Save	F2	Salva il contenuto della finestra di editor attiva in un file con il nome riportato sulla stessa finestra
S**a**ve as...	-	Salva il contenuto della finestra di editor attiva in un file con un nome specificato dall'utente
Change dir...	-	Cambia la directory attuale dell' MS-DOS
Dos shell	-	Esce momentaneamente dal programma per l'esecuzione di applicazioni MS-DOS. Il **GET80** rimane in memoria e non deve essere quindi rilanciato
E**x**it	Alt+X	Esce definitivamente dal programma e ritorna all'MS-DOS.

Se durante l'apertura di un file con l'opzione "Fast Open", si ritrovano nella finestra di editor dei caratteri non desiderati, provvedere a chiudere il file, e quindi riaprire lo stesso con l'opzione "Open"

Menù Edit

Opzione	*Tasto*	*Funzione*
Undo	-	Elimina, se possibile, l'azione dell'ultima opzione eseguita
Cu**t**	Shift+Del	Cancella dalla finestra di editor attiva la parte precedentemente selezionata,mantenendola nella clipboard
Copy	Ctrl+Ins	Copia dalla finestra di editor attiva alla clipboard, la parte precedentemente selezionata
Paste	Shift+Ins	Copia il contenuto della clipboard nella finestra di editor attiva a partire dalla posizione attuale del cursore
Cl**e**ar	Ctrl+Del	Cancella dalla finestra di editor attiva la parte precedentemente selezionata, senza mantenerla nella clipboard
Show clipboard	-	Visualizza la clipboard, con il relativo contenuto

Menù Search

Opzione	*Tasto*	*Funzione*
Find...	-	Cerca una stringa all'interno della finestra di editor attiva
Replace...	-	Cerca una stringa all'interno della finestra di editor attiva e la sostituisce con una seconda stringa
Search again	Alt+A	Ripete l'ultima opzione di "Find" o "Replace" eseguita

Menù Windows

Opzione	*Tasto*	*Funzione*
Tile	-	Rappresenta tutte le finestre di editor aperte distribuendole sull'area di rappresentazione del monitor
C**a**scade	-	Rappresenta tutte le finestre di editor aperte con una sovrapposizione parziale che ne lascia in rappresentazione solo la cornice ed il nome
Size/Move	Ctrl+F5	Consente di muovere e/o ridimensionare la finestra di editor attiva
Zoom	F5	Ridimensiona la finestra di editor attiva alle dimensioni massime
Next	F6	Sposta la finestra di editor attiva sulla sucessiva finestra aperta

Previous	Shift+F6	Sposta la finestra di editor attiva sulla precedente finestra aperta
Close	Alt+F3	Chiude la finestra di editor attiva

Menù Options

Opzione	*Tasto*	*Funzione*
Terminal	Alt+T	Esegue il programma di emulazione terminale intelligente. Per maggiori informazioni su questa opzione, si veda il paragrafo "Emulazione terminale".
Serial Port...	-	Seleziona il baud rate e la linea seriale di comunicazione per il P.C.
Video	-	Seleziona modalità di rappresentazione a colori o in bianco e nero
Information...	-	Visualizza la finestra di informazioni del programma **GET80**

Menù Utility

Opzione	*Tasto*	*Funzione*
GROM	-	Permette di creare l'immagine binaria del file da memorizzare in EPROM, con una ROM DISK utente.
String **E**ditor...	-	Permette di inserire una serie di stringhe definite dall'utente, da utilizzare nella fase di sviluppo del programma applicativo.
Save strings	-	Effettua il salvataggio delle stringhe utente già inserite in un file.

Nella descrizione delle opzioni sopra riportata è stato indicata in grassetto la lettera che consente la selezione veloce dell'opzione una volta attivato il relativo menù a tendina, senza dover utilizzare i tasti freccia, mentre con l'indicazione "*Tasto*" si indica il tasto o la combinazione di tasti, con cui si seleziona l'opzione senza addirittura attivare il menù a tendina. L"indicazione dei "..." che segue alcune opzioni, indica che una volta effettuata la scelta, l'utente dovrà inserire altre informazioni opportunamente richieste dallo stesso programma (es. nome del file, stringa da cercare, la directory da selezionare, ecc.) Una descrizione più approfondita delle modalità d'uso delle opzioni del **GET80** non viene quì riportata, in quanto é demandata direttamente ai numerosi messaggi d'aiuto disponibili nel programma.
Se si fà uso del mouse queste funzioni sono notevolmente semplificate, in quanto é sufficiente fare click sull'opzione del menù desiderata, senza premere alcun tasto da tastiera.

EMULAZIONE TERMINALE

La modalità di emulazione terminale intelligente, gestisce tutte le risorse del P.C., rendendole disponibili alla scheda remota con cui si lavora. Con questo programma è quindi possibile utilizzare i floppy disk, l'hard disk, la stampante, la tastiera ed il monitor del P.C. direttamente con i programmi ed il software in esecuzione sulla scheda remota tramite il sistema operativo **GDOS**.
La comunicazione con la scheda remota viene gestita tramite una delle linee seriali del P.C. (COM) con il collegamento standard indicato nel paragrafo "CAVO DI COMUNICAZIONE SERIALE" e con il protocollo descritto nel capitolo "CARATTERISTICHE TECNICHE DEL **GDOS**".
Una volta attivata l'opzione"Terminal" dal menù "Options" del **GET80**, viene settata la linea seriale attuale con il baud rate attuale e quindi rappresentata una finestra pulita con il cursore in alto a sinistra ed una linea di stato sulla riga 25. Con linea seriale attuale e baud rate attuale s'intendono i valori settati dall'utente (tramite l'opzione "Serial port" od i tasti funzioneF7, F8 da emulazione terminale)

che inizialmente coincidono con i valori di default definiti in fase d'installazione del **GET80.EXE**. Al fine di velocizzare l'uso del'emulazione terminale del **GET80** é conveniente che l'utente provveda ad installarlo od a reinstallarlo prima di farne uso.
Tutti i caratteri trasmessi dalla scheda remota vengono rappresentati sul monitor del P.C., mentre i tasti premuti sulla tastiera del P.C. vengono trasmessi alla scheda remota sfruttando un apposito protocollo logico con cui viene gestita anche la comunicazione di files. Questo protocollo logico é completamente trasparente per l'utente e fà uso degli handshake hardware per regolamentare le comunicazioni veloci. La gestione delle linee di handshake é associata ad un flag salvato nell'**IOBYTE EXTENSION** (vedere apposito paragrafo) con la seguente corrispondenza:

IOBYTE EXTENSION.7 = 1 -> gestione handshake abilitata
IOBYTE EXTENSION.7 = 0 -> gestione handshake disabilitata

Durante la comunicazione con il **GET80**, la gestione degli handshake deve essere abilitata e per questa ragione di default il **GDOS** setta tale flag. Nel caso in cui il programma applicativo realizzato dall'utente non possa utilizzare questi handshake (se ad esempio usa la linea seriale A per comunicare con altri sistemi come modem, altri sistemi, terminali, ecc.) sarà lo stesso programma che si deve preoccupare di resettare il flag degli handshake.
Tutti i programmi realizzati con il **GDOS** possono utilizzare le possibilità dell'emulazione terminale descritte nei paragrafi seguenti, ottenendo così notevoli facilitazioni nella parte di programma che riguarda l'interfaccia utente, mentre la maggioranza dei linguaggi di programmazione viene già fornita con una configurazione adatta a queste caratteristiche.
Il programma di emulazione terminale intelligente opera in modalità completamente asincrona rispetto alla scheda remota che esegue il **GDOS**; quindi non deve essere rispettata nessuna sequenza di accensione o di spegnimento dei due sistemi.

N.B.
Dal sistema operativo MS-DOS, é possibile eseguire il GET80 ed entrare direttamente nella modalità di emulazione terminale intelligente; la sintassi da utilizzare per ottenere questo risultato é la seguente:

GET80 /t<cr>

Comandi dell'emulazione terminale

In emulazione terminale sono disponibili una serie di comandi che aiutano l'utente nell'utilizzo del pacchetto **GDOS 80**. Questi vengono riconosciuti e gestiti tramite vari tasti funzione, come descritto in seguito:

F1 = *Lista dei comandi disponibili*
Visualizza sulla linea di stato inferiore la lista di tasti funzione utilizzabili con la relativa attribuzione; una volta visualizzato questo help provvedere a ritornare in emulazione terminale premendo un qualsiasi tasto. In questa modalità di help i tasti funzione sono visti come tasti generici, quindi non attivano la funzione associata.

F5 = *Impostazione directory corrente*
Consente di variare la directory attuale dell'MS-DOS (come con l'opzione "Change dir...").

F7 = *Selezione linea seriale*
Consente di selezionare la linea seriale da utilizzare sul P.C. per comunicare con la scheda remota (come con l'opzione "Serial port").

F8 = *Selezione Baud Rate*
Consente di selezionare la velocità di comunicazione della linea seriale di comunicazione del P.C. (come con l'opzione "Serial port...").

F10 = *Uscita dall'emulazione terminale*
Esce dalla modalità di emulazione terminale intelligente ritornando alla finestra di editor del **GET80**.

Ctrl Fx = *Mostra stringa utente numero x*
Mostra il teso della stringa utente numero x, sulla linea di stato inferiore ed ha una funzione di promemoria.

Alt Fx = *Trasmette stringa utente numero x*
Trasmette sulla linea seriale di comunicazione del P.C. la stringa utente numero x. A tutti gli effetti questo comando coincide con l'operazione di battitura della stringa sulla tastiera del P.C. e quindi riduce e velocizza tale operazione.

Durante l'esecuzione di uno qualsiasi di questi comandi, solo la linea di stato inferiore é utilizzata per lo scambio di informazioni tra utente e programma, in modo da salvaguardare sempre quello che é l'attuale stato di rappresentazione sul monitor del P.C.

Sequenze di controllo dell'emulazione terminale

Il programma **GET80.EXE** in esecuzione su P.C. riconosce alcune delle sequenze di comando ricevute dalla linea seriale selezionata, caratteristiche del terminale ADDS Viewpoint. Tali sequenze sono riportate nella seguente tabella:

COMANDO	CODICE ASCII	CODICE BYTE
Indirizzamento cursore	**ESC, Y, riga, colonna**	**27, 89, riga, colonna**
Impostazione attributo	**ESC, 0, attributo**	**27, 48, attributo**
Cancellazione fino a fine linea	**ESC, K**	**27, 75**
Cancellazione fino a fine pagina	**ESC, k**	**27, 107**
Cancellazione schermo	**FF**	**12**
Set dell'attributo	**SO**	**14**
Reset dell'attributo	**SI**	**15**

FIGURA 5: SEQUENZE DI CONTROLLO DEL GET80

I valori di riga e colonna possono variare rispettivamente tra 0÷23 e 0÷79 e devono essere forniti con un offset di 32. Quindi se ad esempio si deve posizionare il cursore alla riga 10, colonna 20 deve essere fornita la sequenza:

27,89,42,52.

Per quanto riguarda gli attributi di rappresentazione gestiti dall'emulazione terminale del **GET80** sono solo un sottoinsieme di quelli dello standard ADDS Viewpoint, in particolare:

attributo	CODICE ASCII	CODICE BYTE
Normale	**@**	**64**
Half intensity	**A**	**65**
Reverse	**P**	**80**

FIGURA 6: ATTRIBUTI DI RAPPRESENTAZIONE GESTITI DAL GET80

Tasti speciali dell'emulazione terminale

Nell'emulazione terminale intelligente sono inoltre codificati i tasti speciali della tastiera del P.C. (tasti freccia, Ins, Del, ecc.) che vengono quindi trasmessi alla scheda remota collegata, con i relativi codici standard per **GDOS**:

TASTO	CODICI	CODICI HEX
Freccia UP	05	05
Freccia DOWN	24	18
Freccia LEFT	04	04
Freccia RIGHT	19	13
Page UP	18	12
Page DOWN	03	03
Home	17,82	11,52
End	17,67	11,43
Insert	22	16
Delete	07	07

FIGURA 7: CODICI DEI TASTI SPECIALI GESTITI DAL GET80

Con questa caratteristica si ottengono notevoli facilitazioni soprattutto quando si utilizzano linguaggi di programmazione con un editor integrato (ad esempio il PASCAL), infatti ci si può spostare all'interno del programma applicativo in modo intuitivo, comodo e veloce.

STRINGHE UTENTE

A partire dalla versione 2.4 di **GET80** é stata aggiunta la gestione delle cosiddette stringhe utente. Le stringhe utente coincidono con un massimo di 10 stringhe di lunghezza massima di 70 caratteri che l'utente può editare in qualsiasi momento e quindi utilizzare in modalità di emulazione terminale come comandi digitati da tastiera. E' conveniente che le stringhe utente siano quelle più frequentemente usate durante lo sviluppo del programma infatti la loro funzione principale é quella di ridurre il numero di battute sulla tastiera del P.C. e quindi anche i tempi di sviluppo del programma applicativo. Le stringhe utente possono essere salvate anche in modo permanente in un file di nome **GET80**.FST che viene creato sulla directory da cui é presente il **GET80**.EXE; in fase di partenza il **GET80**.EXE verifica l'esistenza del file **GET80**.FST e se presente provvede a caricare le stringhe utente in un area volatile di memoria, in modo da renderle disponibili per tutte le operazioni. L'utente può utilizzare le stringhe salvate in questa area volatile ricordando che le eventuali modifiche vengono salvate in modo permanente sul file, solo a seguito di un'apposito comando di salvataggio; se questo non viene fornito e si esce dal programma le variazioni effettuate sulle stringhe utente sono definitivamente perse.
La gestione delle stringhe utente é disponibile in tre parti del programma **GET80**, come di seguito descritto:

Menù: Utility - Opzione: Strings Editor
Selezionando questa opzione, viene presentata una videata di editor per le 10 stringhe utente in cui sono presenti 10 caselle testo per l'inserimento delle stringhe, 10 caselle booleane per l'attivazione del carriage return al seguito di ogni stringa, un tasto "**O**K" per chiudere la finestra di editor salvando le stringhe inserite nell'area volatile, un tasto "**S**ave" per salvare in modo permanente le stringhe utente su file ed un tasto "Cancel" per chiudere la finestra di editor senza alcun salvataggio delle stringhe inserite. Le caselle booleane per l'attivazione del carriage return devono essere utilizzate per indicare se la stringa utente deve o non deve essere terminata dal CR, quando viene trasmessa in modalità di emulazione terminale.

Menù: Utility - Opzione: Save Strings
Selezionando questa opzione, viene effettuato un salvataggio delle stringhe utente attualmente salvate nell'area volatile di memoria in quella permanente su file (come il tasto "Save" della precedente opzione). Al termine dell'operazione di salvataggio viene presentata una finestra che riporta l'esito dell'operazione svolta.

Emulazione terminale
In questa fase il **GET80** gestisce le stringhe utente in due modalità; la prima é usata come promemoria per l'utente, infatti visualizza le stringhe salvate nell'area volatile. E' attivata premendo contemporaneamente i tasti "Ctrl" e "Fx" (dove x coincide con il numero di stringa utente da visualizzare), rappresenta la stringa x sulla riga di stato inferiore ed attende un tasto per ritornare in emulazione terminale. La seconda é usata per trasmettere alla scheda remota le stringhe salvate nell'area volatile. E' attivata premendo contemporaneamente i tasti "Alt" e "Fx" (dove x coincide con il numero di stringa utente da rasmettere) e quindi la stringa x é trasmessa così come salvata assieme all'eventuale codice di carriage return. A tutti gli effetti il comando di trasmissione di una stringa utente, attivata con la pressione di soli due tasti, equivale alla battitura sulla tastiera di tutti i caratteri che la compongono.

Una delle operatività più diffuse quando si lavora con il **GDOS** é quella di editare il programma con l'editor del **GET80**, entrare in emulazione terminale e qui dal linguaggio di programmazione in uso,

caricare il programma, trasformarlo, eseguirlo, compilarlo, ecc. Visto che ognuna di queste operazioni richiede la battitura di molti tasti sulla tastiera del P.C. e che durante la fase di sviluppo, queste operazioni vengono ripetute decine di volte, fino a quando il programma applicativo non é completamente testato, si ricava quale sia l'effettiva utilità delle stringhe utente. Se ad esempio si usa lo ZBASIC per la generazione del programma applicativo PRGAPP.ZBA, le stringhe utente potranno essere compilate come segue:

String 1 = N:ZBASIC	con CR attivo
String 2 = LOAD* C:PRGAPP.ZBA	con CR attivo
String 3 = SAVE C:PRGAPP.ZBT	con CR attivo
String 4 = NEW	con CR attivo
String 5 = RUN C:PRGAPP.ZBT	con CR attivo
ecc.	

GROM

A partire dalla versione 2.4 di **GET80** é stata aggiunta la gestione della creazione dell'immagine della EPROM di **GDOS** con l'applicativo sviluppato dall'utente. Per maggiori informazioni a riguardo di questa modalità, si faccia riferimento al paragrafo "GROM: PROGRAMMAZIONE EPROM".

DOS2GDOS: PROGRAMMA DI TRASFORMAZIONE FILE PER P.C.

Il programma **DOS2GDOS** presente nella root del disco di lavoro é un programma da eseguire sul P.C. che trasforma un generico file in formato MS-DOS nell'equivalente formato **GDOS**. Visto che la lunghezza del settore logico dei files per il **GDOS** é di 128 bytes, mentre nell'MS-DOS é di un byte, affinché il **GDOS** possa utilizzare un file MS-DOS si devono aggiungere tanti codici di "end of file" alla fine di questo, fino a quando lo stesso file assume una lunghezza che sia un multiplo di 128 byte. Il programma DOS2GDOS svolge questo lavoro e lo si lancia con la sintassi:

DOS2GDOS <nome>.<est><CR>

Ed il file <nome>.<est> presente nella directory attuale viene ampliato fino a raggiungere una lunghezza che sia multiplo di 128 bytes.
Tra gli usi più frequenti di questo programma si ricordano tutti i casi in cui files che devono essere utilizzati dal **GDOS**, sono creati con specifici programmi MS-DOS (ad esempio editor che possono gestire più di 64 Kbytes, database, ecc.).
L'Editor del **GET80** provvede a salvare i files già con il formato del **GDOS** e per questo motivo il **DOS2GDOS** non deve essere utilizzato sui file da lui creati.

WATCH DOG

Per tutte le schede remote provviste di circuiteria di watch dog, il **GDOS** provvede ad effettuare un'operazione di retrigger durante tutte le operazioni di lettura di files. Questa prerogativa consente di sviluppare applicazioni che facciano uso di tecniche di chain od overlay (con cui il programma in esecuzione provvede a caricare ed a lanciare altri programmi) ed allo stesso tempo utilizzano la circuiteria di watch dog. Da questo si ricava che é normale non vedere più attivarsi il LED del watch dog, quando si effettua un'operazione di lettura files. Il retrigger della circuiteria viene effettuato in corrispondenza della lettura di ogni settore logico, quindi il watch dog non si attiverà quando si leggono files da ROM DISK o RAM DISK in cui il tempo di accesso é bassissimo, mentre quando si leggono files dai drive del P.C. l'intervento é subordinato al tempo di accesso che varia a seconda del calcolatore.

COMANDI DIRETTI DEL GDOS

Il sistema operativo **GDOS** in esecuzione sulla scheda remota è in grado di riconoscere e gestire una serie di comandi diretti che aumentano la versatilità di tutto il sistema; questi comandi possono essere forniti quando il **GDOS** si trova in modalità comandi, evidenziata dalla presenza del prompt **<d>:ABACO**°**>** a capo linea, dove d coincide con il drive attuale del **GDOS**.
Vengono di seguito descritti tali comandi diretti, proponendo con esempi la corretta sintassi degli stessi. In particolare la descrizione sucessiva è basata anche sull'uso del programma per il P.C. descritto nel precedente paragrafo.

SAVE nn d:nome.est Salva il contenuto della memoria programma TPA nel file nome.est che viene creato sul drive d. Con nn deve essere indicata la lunghezza in blocchi da 256 byte dell'area di memoria da salvare, in hex. Il comando non effettua controlli sulla presenza del file da creare, quindi nel caso esista un file con lo stesso nome, questo sarà riscritto. Per salvare i primi 5 K della memoria in un file di nome PROVA.G80 da creare sul drive A del P.C. si deve fornire il comando:

SAVE 14 A:PROVA.G80<cr>

d:nome Carica il contenuto del command file nome.G80 presente sul drive d nella memoria TPA della scheda remota per poi eseguirlo. Il comando carica ed esegue il programma nel caso il file esista, viceversa non effettua alcuna operazione e viene ripresentato il prompt. Da notare che se il command file esiste viene settato il drive attuale del **GDOS** con quello da cui é stato lanciato il porogramma. Per eseguire il programma presente nel file NSB8.G80 presente sul drive C del P.C. si deve fornire il comando:

C:NSB8<cr>

VER Visualizza sul dispositivo di consolle un messaggio che riporta tutte le informazioni riguardanti la versione del sistema operativo **GDOS** in esecuzione sulla scheda remota. Una volta rappresentata tale linea di stato, il sistema si riporta sotto il controllo del sistema operativo notificandolo con la ricomparsa del prompt d:ABACO°>

RIT Rilancia il programma presente nell'area TPA del sistema, senza modificare il precedente contenuto. In altri termini viene effettuato un salto assoluto alla locazione 0100H, eseguendo quanto c'è in RAM. Tale comando è molto utile se ad esempio utilizzato assieme a linguaggi di programmazione, infatti in caso di uscita accidentale da quest'ultimi è possibile rientrare senza perdere quanto già sviluppato. Da notare inoltre che questo comando può essere utilizzato solo ed esclusivamente se l'ultimo programma in esecuzione ha lasciato in memoria una immagine corretta per una nuova esecuzione (come NSB8, ecc.).

I comandi diretti del **GDOS** devono rigorosamente seguire la sintassi sopra riportata. In caso contrario con questa versione si possono presentare funzionamenti non corretti.

PROGRAMMI DI UTILITY PER GDOS

Il sistema operativo **GDOS** in esecuzione sulla scheda remota è in grado di eseguire una serie di programmi di utility che facilitano l'utilizzo di tutto il sistema. La maggioranza di questi programmi riguarda la gestione di files e quindi di tutte le memorie di massa. Anche questi programmi possono essere forniti quando il **GDOS** si trova in modalità comandi, evidenziata dalla presenza del prompt **<d>:ABACO°>** a capo linea.
Vengono di seguito descritti tali programmi di utility, proponendo con esempi la corretta sintassi degli stessi. In particolare la descrizione sucessiva è basata anche sull'uso del programma per il P.C. descritto in uno dei precedenti paragrafi.

d1:REN d2:nome1.est1=nome2.est2 Cambia il none del file nome1.est1 presente sul drive d2, assegnandogli il nome nuovo nome2.est2. Il programma esegue l'operazione di rename solo nel caso in cui il file nome1.est1 esista sul drive d, ed una volta completata l'operazione ripresenta il prompt; se invece il file non esiste, non viene effettuata alcuna operazione e quindi ricompare immediatamente il prompt. L'indicazione del drive d1 riguarda il drive in cui è memorizzato il command file REN.G80. Se ad esempio il file PROVA.G80 presente sulla Ram Disk della scheda remota deve essere chiamato con il nome di AUTORUN.G80 ed il programma REN.G80 è allocato sul drive C del PC, si deve fornire il seguente comando:

C:REN M:PROVA.G80=AUTORUN.G80<cr>

d1:DIR d2: Presenta sul monitor l'elenco dei file presenti sul drive d2. I nomi e le estensioni dei files sono riportate incolonnati a sinistra, separati da uno spazio ed a completamento della pagina,la visualizzazione si interrompe per riprendere su pressione di un tasto, come indicato da un apposito messaggio. L'indicazione del drive d1 riguarda il drive in cui è memorizzato il command file DIR.G80. Le indicazioni dei drive d1 e d2 coincidono con i nomi logici dei drive, quindi per avere ad esempio il direttorio del drive A, quando il programma DIR.G80 è allocato sul drive C, si deve fornire il comando:

C:DIR A:<cr>

Con tale comando si visualizza quindi il contenuto del drive A del P.C., caricando il programma dal drive C.

d1:ERA d2:nome.est Cancella il file nome.est dal drive d2. Il programma verifica la presenza del file e nel caso sia presente lo cancella per poi ripresentare il prompt del sistema operativo, viceversa non fornisce alcun messaggio d'errore e si limita a ripresentare il prompt. La cancellazione effettuata è definitiva e quindi prima di utilizare questo programma si deve verificare con attenzione la sua idoneità. L'indicazione del drive d1 riguarda il drive in cui è memorizzato il command file ERA.G80. Per cancellare ad esempio il file PROVA.B dal drive A del P.C., quando il programma ERA.G80 è allocato in Rom Disk, si deve fornire il comando:

N:ERA A:PROVA.B<cr>

d1:FORMAT d2: Effettua la formmattazione del drive d2. Il programma verifica che il drive da formattare sia un drive di Ram Disk ed in caso negativo termina ripresentando il prompt, viceversa prosegue chiedendo una conferma all'utente. La formattazione effettuata è definitiva e quindi prima di utilizare questo programma si deve verificare con attenzione la sua idoneità. L'indicazione del drive d1 riguarda il drive in cui è memorizzato il command file FORMAT.G80. Per formattare il drive di Ram Disk M di una scheda remota, quando ad esempio il programma FORMAT.G80 è allocato sul drive A del P.C., si deve fornire il comando:

A:FORM80F M:

d1:COPY d2:nome.est=d3: Effettua la copia del file nome.est presente sul drive d2 che viene posto sul drive d3. Il programma verifica la presenza del file sul drive d2 e se presente prosegue con la copia, viceversa termina ripresentando il prompt. Durante l'operazione di copia vengono effettuati alcuni controlli come ad esempio quello di spazio ancora disponibile sul drive d3. L'indicazione del drive d1 riguarda il drive in cui è memorizzato il command file COPY.G80. Se ad esempio si deve copiare il file NSB8.G80 presente sul drive A del P.C. sul drive M di Ram Disk ed il programma COPY.G80 è allocato in Rom Disk, si deve fornire il comando:

N:COPY A:NSB8.G80=M:<cr>

Per tutti i programmi esterni sopra descritti (ERA, DIR, REN, COPY) non possono essere utilizzate le wild card, caratteristiche degli altri sistemi operativi. Da ricordare inoltre che i programmi di utility coincidono con dei veri e propri command file per **GDOS** che effettuano le operazioni descritte e che possono essere sostituiti anche da programmi utente realizzati con un generico linguaggio di programmazione.
Al fine di velocizzare l'esecuzione dei programmi di utility esterni, questi vengono sempre forniti memorizzati in Rom Disk.

PROCEDURE DI UTILITY PER GDOS

Il sistema operativo **GDOS** dispone di una serie di procedure di utilizzo generale che consentono di utilizzare alcune risorse di bordo ad alto livello; tali procedure possono essere direttamente chiamate dai linguaggi di programmazione e quindi direttamente usate dall'utente. Tutte le procedure sono chiamate direttamente da programma, specificandone l'indirizzo di allocazione; tale indirizzo non varia al variare della versione di **GDOS** e del tipo di scheda remota quindi é sempre mantenuta la compatibilità softwaree la trasportabilità del codice. La seguente tabella riporta gli indirizzi di allocazione di queste procedure di utility descritte nei paragrafi sucessivi:

PROCEDURA UTILITY	INDIRIZZO	INDIRIZZO HEX
Format drive M di RAM DISK	**62003**	**F233**
Format drive O di RAM DISK	**62006**	**F236**
Lettura, scrittura EEPROM seriale	**62009**	**F239**
Format drive L di RAM DISK	**62012**	**F23C**
Attivazione LEDs interfaccia operatore locale	**62015**	**F23F**

FIGURA 8: INDIRIZZI PROCEDURE DI UTILITY

FORMAT DRIVE DI RAM DISK

Con queste procedure l'utente ha la possibilità di formmattare tutti i drive di RAM DISK della scheda remota (compreso il drive su RAM CARD esterno), direttamente dal programma applicativo. La procedura non richiede parametri e non ne restituisce, la si chiama semplicemente all'indirizzo di allocazione riportato in figura 8 come una normale routine in linguaggio macchina; é assoluta e lascia inalterata tutta la memoria di lavoro TPA e tutti i registri del microprocessore. Per quanto riguarda la corrispondenza tra i nomi dei drive (L,M,O) ed i relativi dispositivi di memoria, fare riferimento al paragrafo "RAM ROM DISK".

LETTURA SCRITTURA EEPROM SERIALE

Con questa procedura si ha la possibilità di leggere e/o scrivere una serie di byte nei confronti della EEPROM seriale della scheda remota, tramite tutti i linguaggi di programmazione disponibili per il sistema operativo **GDOS**. Per l'utilizzo di tale procedura si devono prima settare i parametri d'ingresso, quindi chiamarla all'indirizzo di allocazione riportato in figura 8 come una normale routine in linguaggio macchina ed infine prelevare i parametri d'uscita. La procedura é assoluta e lascia inalterata tutta la memoria di lavoro TPA e tutti i registri del microprocessore. Per quanto riguarda il passaggio dei parametri é stata scelta l'area di memoria che coincide con il BUFFER del **GDOS** in modo da non incorrere nel problema di modifica di aree di memoria già utilizzate. In questo buffer i parametri sono così organizzati:

Indirizzo		*Parametro*
0080H = 128	->	Selettore del tipo di operazione: 00H -> Lettura FFH -> Scrittura
0081H = 129	->	Numero di byte da leggere o scrivere (<=123)
0082H = 130	->	Byte low dell'indirizzo d'inizio del blocco dati in EEPROM
0083H = 131	->	Byte high dell'indirizzo d'inizio del blocco dati in EEPROM
0084H = 132	->	Risultato dell'operazione svolta: 00H -> Operazione riuscita correttamente 01H -> EEPROM non presente sulla scheda 02H -> Parametri d'ingresso non validi 04H -> Sequenza anormale di accesso all'EEPROM
0085H = 133	->	Primo byte da scrivere o letto
0086H = 134	->	Secondo byte da scrivere o letto
: : :	:	: : : : :
: : :	:	: : : : :
00FEH = 254	->	Centoventiduesimo byte da scrivere o letto
00FFH = 255	->	Centoventitreesimo byte da scrivere o letto

Il settaggio dei parametri d'ingresso ed il prelevamento dei parametri d'uscita deve essere effettuato con le apposite istruzioni di accesso alla memoria, presenti nel linguaggio di programmazione in uso. La procedura illustrata verifica la validità dei parametri d'ingresso e nel caso non siano congruenti lo segnala nell'apposito parametro di uscita che riporta il risultato dell'operazione. Per questa verifica la procedura tiene conto anche delle dimensioni della EEPROM montata, verificando che gli indirizzi interessati dall'operazione di lettura o scrittura non siano fuori range. Affinché la EEPROM seriale sia gestita correttamente in tutto il suo campo d'indirizzamento, deve essere opportunamente inizializzata; la procedura di inizializzazione può essere effettuata solo dalla **Grifo®** e provvede a salvare informazioni indispensabili al **GDOS**. Il dispositivo inizializzato é contraddistinto da un pallino colorato. La procedura di gestione della EEPROM non consente un'accesso a tutto lo spazio d'indirizzamento del dispositivo. Infatti i primi 32 bytes di questo sono riservati per uso interno e di configurazione e non quindi accessibili all'utente. Il **GDOS** gestisce cinque size di EEPROM seriali che sono: 2 KBit (256 Bytes), 4 KBit (512 Bytes), 16 KBit (2048 Bytes), 32 KBit (4096 Bytes) e 64 KBit (8192 Bytes); la verifica del size e delle altre informazioni di configurazione é effettuata solo alla partenza del **GDOS** sulla scheda remota, ovvero dopo un reset od un power on di quest'ultima.

ATTIVAZIONE LEDS INTERFACCIA OPERATORE LOCALE

Con questa procedura si ha la possibilità di attivare e/o disattivare i LEDs dell'interfaccia operatore locale collegata a 16 linee di I/O della scheda remota (si veda paragrafo "INTERFACCIA OPERATORE LOCALE") tramite tutti i linguaggi di programmazione disponibili per il sistema operativo **GDOS**. Per l'utilizzo di tale procedura si devono prima settare i parametri d'ingresso, quindi chiamarla all'indirizzo di allocazione riportato in figura 8 come una normale routine in linguaggio macchina. La procedura é assoluta e lascia inalterata tutta la memoria di lavoro TPA e tutti i registri del microprocessore. Per quanto riguarda il passaggio dei parametri é stata scelta l'area di memoria che coincide con il BUFFER del **GDOS** in modo da non incorrere nel problema di modifica di aree di memoria già utilizzate. In questo buffer i parametri sono così organizzati:

Indirizzo		*Parametro*
0080H = 128	->	Stato dei LEDs 1÷8 dell'interfaccia operatore locale
0081H = 129	->	Stato dei LEDs 9÷16 dell'interfaccia operatore locale

Il settaggio dei parametri d'ingresso ed il prelevamento dei parametri d'uscita deve essere effettuato con le apposite istruzioni di accesso alla memoria, presenti nel linguaggio di programmazione in uso. Lo stato dei 16 LEDs dell'interfaccia operatore locale é associato ai 16 bit dei due parametri d'ingresso con la seguente corrispondenza:

Bit 0 del Byte salvato in 080H	->	Stato LED1
Bit 1 del Byte salvato in 080H	->	Stato LED2
: : : : : :	:	: :
Bit 7 del Byte salvato in 080H	->	Stato LED8
Bit 0 del Byte salvato in 081H	->	Stato LED9
: : : : : :	:	: :
Bit 7 del Byte salvato in 081H	->	Stato LED16

Per lo stato dei LEDs vale invece la seguente corrispondenza:

Bit a 0	->	LED spento
Bit a 1	->	LED acceso

L'utente deve prestare particolare attenzione alla chiamata di questa procedura di utility ed in particolare deve preventivamente verificare che alla scheda remota sia collegata l'interfaccia operatore locale e non una altra interfaccia hardware. Per la gestione dell'interfaccia operatore locale sono infatti gestite le linee di I/O che con la chiamata a questa procedura vengono variate, con conseguenti effetti imprevedibili nel caso che a tali linee sia collegato un hardware non corretto.
Viene di seguito riportato la disposizione dei LEDs gestiti da questa procedura nel caso dell'interfaccia operatore **QTP 24P**.

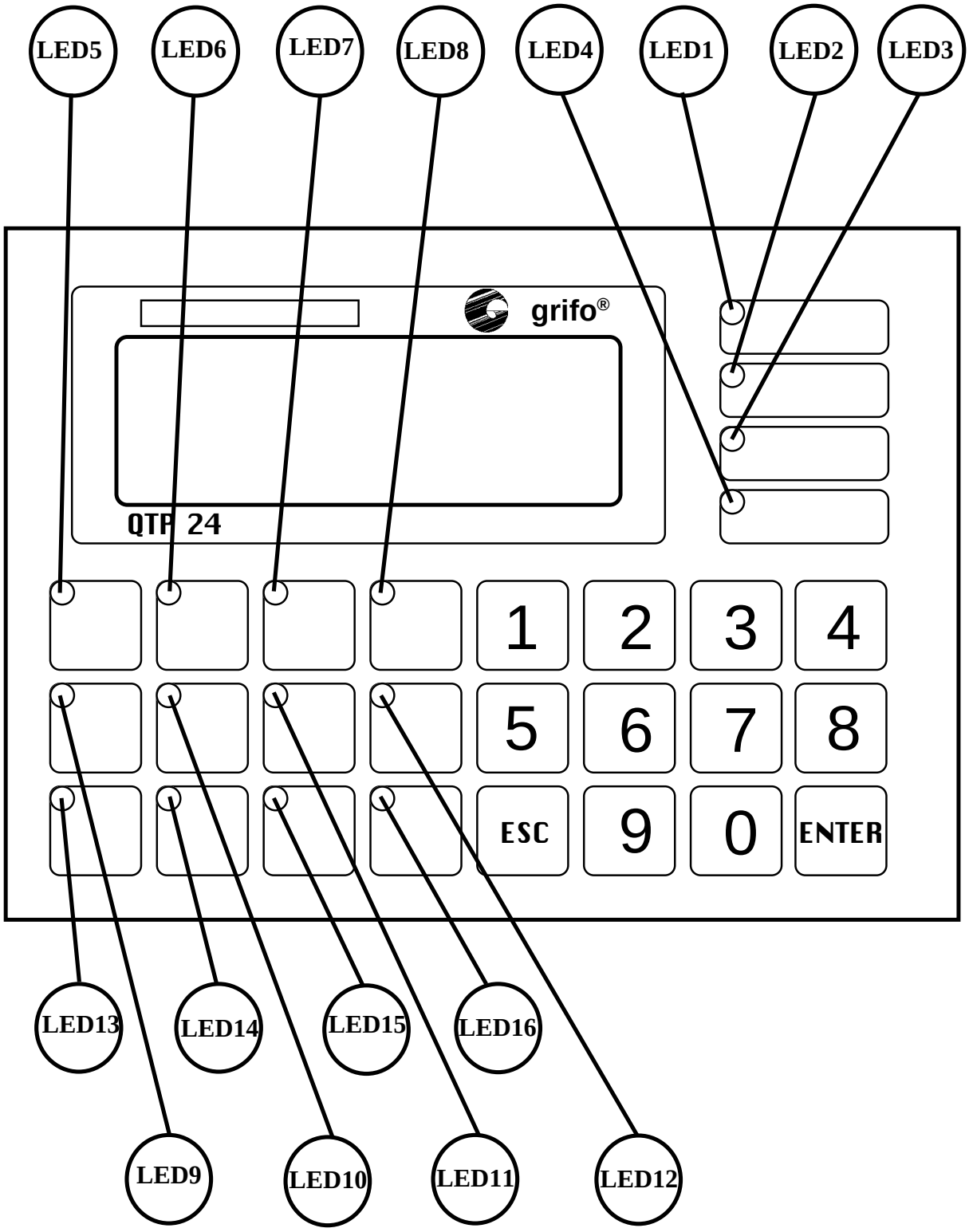


FIGURA 9: DISPOSIZIONE LEDS SU QTP 24P

RAM ROM DISK

Il sistema operativo locale **GDOS** si preoccupa anche di gestire la memoria eccedente i 64 K RAM ed i 16K EPROM o FLASH EPROM, rispettivamente come RAM Disk e ROM Disk. In questo modo tutti i file gestibili da sistema operativo, possono essere scritti e letti anche sulle risorse di memoria di massa locali.
Questa implementazione consente di accellerare notevolmente tutte quelle che sono le operazioni di caricamento e salvataggio dati, con un grosso risparmio di tempo sopratutto per quanto riguarda lo sviluppo software di tutta l'applicazione.
In particolare è disponibile:

RAM Disk a bordo scheda remota, vista come drive 13 = drive logico M
ROM Disk a bordo scheda remota, vista come drive 14 = drive logico N
RAM Disk a bordo scheda remota, vista come drive 15 = drive logico O (ausiliaria)
RAM Disk su RAM Card esterna, vista come drive 12 = drive logico L

Quindi se ad esempio si dovesse salvare il contenuto dei primi 10 K della memoria TPA tramite il sistema operativo locale, si può fornire il comando diretto:

<d>:ABACO°>**SAVE 28 M:nome.est**<cr>
con cui si crea il file nome.est nella RAM disk locale. Supponendo invece di avere in Rom disk il command file NSB8.G80, per eseguirlo è sufficiente fornire il comando diretto:

<d>:ABACO°>**N:NSB8**<cr>

Da notare che sia le dimensioni delle RAM/ROM disk, sia l'effettiva presenza di RAM/ROM disk è in relazione al tipo di scheda remota che esegue il sistema operativo **GDOS**. Infatti ogni tipo di scheda remota ha delle dimensioni minime e massime variabili per ogni tipo di drive, come indicato nella seguente tabella:

SCHEDA REMOTA	DRIVE L	DRIVE M	DRIVE N	DRIVE O
GPC° 80F	**0,5÷16 MByte**	**0÷188 KByte**	**109÷237 KByte**	**assente**
GPC° 81F	**0,5÷16 MByte**	**assente**	**109÷493 KByte**	**assente**
GPC° 011	**0,5÷16 MByte**	**0÷128 KByte**	**45÷237 KBYte**	**0÷58 KByte**
GPC° 15A	**0,5÷16 MByte**	**61 KByte**	**109÷493 KByte**	**assente**
GPC° 15R	**0,5÷16 MByte**	**61÷445 KByte**	**109÷493 KByte**	**assente**
GPC° 153	**0,5÷16 MByte**	**61÷445 KByte**	**109÷493 KByte**	**assente**
GPC° 183	**0,5÷16 MByte**	**61÷445 KByte**	**109÷493 KByte**	**assente**

FIGURA 10: DIMENSIONI DRIVE LOCALI

Quindi prima di utilizzare la ROM disk o la RAM disk, si deve verificare il tipo di scheda remota che si stà utilizzando e la configurazione di tale scheda in termini di memorie. Nel caso in cui le risorse di memorie locali non siano sufficienti, si può far uso di dispositivi ausiliari di memorie di massa come la **MCI 64**. Infatti la versione **MCI** del **GDOS** é in grado di gestire un drive di RAM disk di grosse dimensioni associato ad una RAM card gestita tramite la **MCI 64** con cui possono essere facilmente risolti problemi di data loghin, trasformazione dati, ecc. (per maggiori informazioni fare riferimento ai paragrafi "INTERFACCIE PER I/O DIGITALI" e "RAM DISK CON MCI 64").
Nelle schede in cui è presente circuiteria di protezione in scrittura della RAM, è sempre conveniente abilitare questa funzione prima di iniziare ad utilizzare la RAM disk; infatti il **GDOS** gestisce opportunamente tale circuiteria, conferendo ai dati memorizzati una sicurezza estrema.
Il **GDOS** in fase di power on o reset verifica la presenza e le dimensioni di RAM e ROM disk, in modo da poter sucessivamente gestire queste sezioni senza nessuna operazione da parte dell'utente. Inoltre durante la verifica di presenza della RAM disk, si occupa anche di effettuare un test di formattazione della stessa. Nel caso in cui la RAM disk non sia formattata (primo utilizzo o mancanza di alimentazione della scheda remota), provvede a farlo per tutto lo spazio disponibile, rendendola quindi sempre utilizzabile. In caso di dubbi a riguardo dello stato della RAM disk, l'utente può sempre utilizzare l'apposito programma di utility FORMAT.G80 che su conferma dell'utente formatta il drive di RAM disk prescelto (si vedano i paragrafi "PROGRAMMI DI UTILITY PER **GDOS**" e "FORMAT DRIVE DI RAM DISK").
Da ricordare che i driver di RAM disk possono essere utilizzati sia per operazioni di scrittura che di lettura, mentre il driver di ROM disk può essere utilizzato solo per operazioni di lettura se è associato ad un dispositivo EPROM, oppure di lettura e scrittura occasionale se associato ad un dispositivo di FLASH EPROM. Per mqggiori informazioni in merito, si faccia riferimento agli appositi paragrafi "PROGRAMMAZIONE EPROM E FLASH EPROM".
L'utente può gestire tutti i drive di memoria di massa, utilizzando le apposite istruzioni del linguaggio di programmazione scelto. Naturalmente i dati possono essere letti e/o scritti sui drive sotto forma di files con le modalità ritenute più opportune dall'utente; tali modalità e la sintassi d'uso delle istruzioni sono da ricercare nella documentazione del linguaggio di programmazione.
Vengono di seguito riportate una serie di informazioni che riguardano i limiti di utilizzo della RAM e ROM disk quando si lavora con il **GDOS**; tali limiti sono esclusivamente dovuti al fatto che i drive attuali sono gestiti in modo sequenziale e possono comunque essere facilmente superati con dei semplici accorgimenti.

1) La scrittura su files, con incremento delle loro dimensioni, può avvenire solo su **un file alla volta** e solo se lo stesso file, si trova in **ultima posizione** della directory ovvero se è stato l'ultimo file creato in ordine temporale. La scrittura su files senza incremento, può avvenire invece, senza limitazioni. Questo limite é facilmente superato se i file di cui aumenteranno le dimensioni, quando vengono creati, vengono anche scritti con quelle che sono le loro massime dimensioni.
2) Le indicazioni di spazio libero sui driver locali, e di stato di un file sono solo indicative e non possono essere utilizzatecome parametro oggettivo nei calcoli di spazio libero od occupato. Per ottenere le informazioni descritte fare riferimento ad appositi contatori gestiti via software, oppure ai risultati delle operazioni effettuate nei confronti del file.
3) I nomi dei files che contengono wild card (* o ?) non sono gestite sui driver locali e quindi non possono essere utilizzate.
4) Su un drive locale non possono essere salvati più di 64 files contemporaneamente.
5) Sul drive di ROM disk (N) non possono essere mantenuti contemporaneamente aperti più di 3 files, mentre sul drive di RAM disk associato ad una RAM card (L) con write protect attivo, non possono essere mantenuti contemporaneamente aperti più di 10 files.

PROGRAMMA IN AUTORUN

Con il sistema operativo **GDOS** implementato sulla scheda remota si ha la possibilità, in fase di reset o power on, di eseguire automaticamente quello che è un programma applicativo generato dall'utente. In questo modo si può verificare anche la funzionalità del sistema nella fase di accensione oppure si può rendere definitiva un'applicazione già completamente testata.
Per effettuare questa operazione è sufficiente memorizzare nella RAM o ROM disk della scheda remota, il file che si desidera provare, con il nome: **AUTORUN.G80**. Il sistema operativo **GDOS** in fase di power on o reset verifica prima la presenza di questo file in RAM/ROM disk (con priorità della ROM disk), quindi se presente lo esegue, viceversa si pone in posizione di attesa comandi sollecitando l'operatore con il prompt N:ABACO°>. Nel caso di schede remote con più drive locali di memorie di massa, la verifica di presenza del file di AUTORUN viene effettuata su tutti i drive con la priorità di seguito riportata: N, M, O, L.
Se il file di AUTORUN viene trovato su uno dei driver della scheda remota, il **GDOS** si occupa di caricarlo in memoria ed eseguirlo, senza trasmettere sulla linea seriale di consolle, quello che è il classico messaggio di power on descritto in precedenza. In questo modo vengono eseguite solo le operazioni di inizializzazione, senza aggiungere alcuna operazione al programma applicativo realizzato dall'utente.
Se l'utente desidera utilizzare il sistema operativo **GDOS**, anche se è presente un file AUTORUN.G80 nella RAM/ROM Disk della scheda remota, deve settare opportunamente la modalità operativa. Questa possibilità é di particolare interesse quando si deve intervenire su un sistema già installato. Disabilitando la verifica dell'autorun l'utente può infatti intervenire sul programma, anche sul campo, e quindi riattivare il tutto senza alcuna complicazione e perdite di tempo. Oppure é possibile verificare gli effetti del programma una volta che questo ha lavorato, effettuare interventi per acquisire dati statistici, ecc. In particolare:

Modalità RUN: Il programma di AUTORUN se presente, viene eseguito in modo automatico, viceversa si entra nella modalità comandi del **GDOS**.

Modalità DEBUG: Anche se presente un programma di AUTORUN, si entra sempre nella modalità comandi del **GDOS**.

Per quanto riguarda la selezione della modalità operativa, fare riferimento al manuale tecnico della scheda remota; normalmente é presente un jumper a tre posizioni il cui settaggio é inoltre visualizzato su due appositi LEDs di stato (LED verde per modalità RUN e LED giallo per modalità DEBUG). Nelle schede sprovviste di questo jumper (vedi **GPC® 80F, GPC® 011**) si deve utilizzare il DIP 8 del Dip Switch di bordo con la seguente corrispondenza:

DIP 8 in OFF: Modalità RUN
DIP 8 in ON: Modalità DEBUG

Da notare che il file a cui viene assegnato il nome di AUTORUN.G80 deve essere un file oggetto eseguibile con origine 0100H (command file del **GDOS**).
In relazione a questo discorso, non esistono problemi quando si lavora con strutture che generano direttamente un command file (vedi compilatori), infatti da quest'ultimi si ottiene direttamente un command file. Viceversa lavorando con strutture diverse (vedi programmi interpretati) questa operazione non è immediata, ma comunque possibile. Per informazioni più approfondite fare riferimento ai specifici manali dei linguaggi di programmazione.

PROGRAMMAZIONE EPROM E FLASH EPROM

L'ultimo passo da effettuare per ottenere l'applicazione definitiva tramite il pacchetto software **GDOS** è la programmazione della EPROM da montare sulla scheda remota o della FLASH EPROM già montata sulla scheda stessa. Sia la EPROM che la FLASH EPROM sono viste dal sistema operativo **GDOS** come ROM disk, e quindi i dati su questi dispositivi sono sempre organizzati sotto forma di files. La EPROM deve essere programmata tramite un apposito programmatore esterno ed una volta realizzata può essere modificata solo tramite una sostituzione od una cancellazione e sucessiva riprogrammazione; in ogni caso é necessario intervenire sull'hardware della scheda per smontare e rimontare la stessa EPROM. Si consiglia quindi di programmare l'applicativo in EPROM solo quando questo é completamente testato. La FLASH EPROM invece offre il vantaggio di poter essere direttamente programmata dalla scheda remota su cui é montata. Quindi nel caso di variazioni od aggiunte, non é necessario smontare e rimontare il componente ma utilizzando semplicemente la linea seriale l'utente può effettuare la programmazione o la riprogrammazione del dispositivo.
In ROM disk, oltre ai programmi applicativi, possono essere salvati anche programmi di supporto (o comunque qualsiasi tipo di file) che possono essere utilizzati anche nella fase di sviluppo, per accellerare i tempi di esecuzione, oppure files di messaggi, configurazione, ecc.

GROM: PROGRAMMAZIONE EPROM

Per la realizzazione della EPROM è stato prevista un'apposita modalità nel programma per P.C. GET80.EXE, come già descritto nei precedenti paragrafi. Questa modalità, in base alle indicazioni dell'utente, prepara un'immagine binaria con cui può essere programmata la EPROM da montare sulla scheda remota. L'utente deve svolgere le seguenti fasi:

a) Assicurarsi che tutti i files da porre in ROM disk, più il sistema operativo **GDOS**, siano presenti nei drive di cui dispone il P.C. Infatti la modalità **GROM** gestisce tutti i drive ed anche le directory di questo, da cui preleva i files da salvare in EPROM, ma non i drive locali della scheda remota; per effettuare questa operazione è sufficiente copiare tutti i files con le modalità descritte in questo manuale o con l'MS-DOS.

b) Eseguire il programma **GET80** sul Personal Computer in esecuzione dell'MS-DOS e con le indicazioni del paragrafo "GET80: PROGRAMMA DI SUPPORTO PER P.C." entrare nell'opzione **GROM**. In questa fase compare una videata a colori che richiede i dati per la creazione della EPROM, con le modalità sotto descritte.

c) La prima operazione da effettuare é la selezione della dimensione della EPROM che deve essere programmata. Il **GROM** controlla che la dimensione del dispositivo scelta sia sufficiente a contenere tutti i files scelti dall'utente ed in caso negativo avvisa con un'apposito messaggio di stato. In ogni caso l'utente in base allo spazio occupato in ROM disk può variare il size prima di procedere con l'effettiva creazione dell'immagine EPROM con una conseguente ottimizzazione dei costi.

d) La seconda operazione da effettuare é la selezione del tipo di scheda in uso. Infatti il **GROM** si presta alla programmazione della EPROM per tutte le schede del carteggio **Grifo®** su cui é disponibile il **GDOS**. A tutti gli effetti la selezione si limita ad un flag booleano da attivare in corrispondenza dell'uso della **GPC® 011** e disattivare per tutte le altre schede.

e) Come terzo passo si deve introdurre il nome del file binario che contiene l'immagine del sistema operativo **GDOS**, per la scheda remota in uso. Tale file è presente nella directory AUTORUN del disco di lavoro e per l'inserimento si deve attivare l'opzione "**G**DOS80 File" a cui segue la presentazione di una classica finestra di dialogo per ricerca files in cui sono disponibili tutte le facilitazioni che consentono di variare directory, cercare determinati tipi di files, usare impostazioni precedenti, ecc. L'introduzione di questo nome file viene riportata a fianco della casella "GDOS80 File".

f) Come quarto passo si deve introdurre il nome del file binario, in cui il **GROM** pone l'immagine della EPROM da creare. Per questo inserimento si deve attivare l'opzione "**B**IN EPROM File" a cui segue la presentazione di una classica finestra di dialogo per ricerca files in cui sono disponibili tutte le facilitazioni che consentono di variare directory, cercare determinati tipi di files, usare impostazioni precedenti, ecc. Il file specificato in questa fase é generato dal **GROM** e la sua dimensione coincide sempre con la dimensione selezionata al punto **c**; l'eventuale area non utilizzata é riempita con dati non significativi (0FFH). L'introduzione di questo nome file viene riportata a fianco della casella "BIN EPROM File".

g) Terminate le fasi descritte, si può entrare nella fase di vera e propria creazione della ROM disk: possono infatti essere inseriti in sequenza i nomi dei files da salvare con i rispettivi nomi che dovranno avere in ROM disk. In particolare una volta attivata l'opzione "**R**OM-Disk" possono essere utilizzati i tasti:

"freccia","Pgxx","End","Home"	-> per spostare la linea che indica il file attualmente scelto;
"INS"	-> per aggiungere un file in ROM disk in ultima posizione;
"ENTER"	-> per modificare il file attualmente scelto;
"DEL"	-> per cancellare il file attualmente scelto;
"Ctrl+Del"	-> per cancellare tutti i file già inseriti.

Durante la fase di aggiunta o modifica di un file, viene prima presentata una finestra di dialogo per ricerca files ed a scelta effettuata una seconda finestra per l'inserimento del nome che tale file dovrà avere in ROM disk; questa prerogativa è molto utile soprattutto perchè fornisce la possibilità di creare il file di AUTORUN in ROM disk, partendo da un files su P.C. con nome diverso. Le modalità di inserzione di questi nomi sono ampiamente descritte dallo stesso programma, che oltretutto fornisce un'indicazione aggiornata del numero di files presenti in ROM disk e dello spazio libero rimasto a disposizione. Quando lo spazio libero (che è in relazione al tipo di EPROM selezionata ed ai files già inseriti) è insufficiente per l'inserimento di un altro file, il **GROM** visualizza un apposito messaggio di stato, come descritto al punto **c**. Anche nel caso di inserimento di un nome file già presente in ROM disk viene presentato un apposito messaggio di stato che avvisa l'utente fornendogli la possibilità di proseguire o abortire. La finestra di rappresentazione dei files inseriti in ROM disk é composta da sole 9 righe ed é a scorrimento, in modo da rendere sempre possibile la gestione dei massimi 64 files inseribili.

h) Con la selezione dell'opzione "**C**ancel" la modalità **GROM** viene abbandonata senza creare alcuna immagine EPROM, mentre con l'opzione "**M**ake" si confermano tutti i dati inseriti ed il **GROM** rappresenta una finestra con una barra a scorrimento che indica la percentuale di lavoro svolto ed una sottostante linea di stato che informa l'utente sull'operazione in corso. E' solo in questa fase che viene effettivamente creato il file con l'immagine binaria della EPROM ed al termine, sempre sulla riga di stato, é riportato un messaggio che riporta l'esito finale di tutta l'operazione. Nel caso di abbandono i dati già inseriti non sono persi, bensì mantenuti in memoria e quindi di nuovo disponibili rientrando nella modalità GROM. Questa possibilità é

particolarmente interessante quando l'utente si rende conto di non avere tutti i file disponibili per la creazione della ROM disk dopo aver già iniziato la sua creazione (ad esempio deve compilare un programma, aggiornare un files dati, ecc.). L'uscita dal programma è comunque sempre possibile tramite la pressione del tasto "ESC".

i) Uscire dal programma GET80 riportandosi nell'ambiente MS-DOS in cui è possibile verificare la presenza del file creato sul drive e l'eventuale directory preselezionate al punto **f**.

j) Predisporre il Personal per la programmazione della EPROM di dimensioni selezionate al punto **c**, effettuando tutte le operazioni necessarie a seconda del programmatore in uso.

k) Programmare il file ottenuto al punto **h** all'interno della EPROM scelta. Tale file deve essere programmato a partire dall'indirizzo 0000H, quindi se il programmatore in uso richiede il formato HEX Intel provvedere a trasformare il file binario descritto nell'equivalente HEX tramite il programma GHEX2.COM, specificando come offset 0000H. La sintassi di utilizzo del GHEX2 in queste modalità è: GHEX2 <nome>.<est><CR>
da fornire sotto il controllo dell'MS-DOS e da cui si ottiene il file <nome>.HEX nel formato HEX Intel.

l) La EPROM ottenuta una volta montata sulla scheda remota, all'atto del reset o del power ON esegue il **GDOS** mettendo a disposizione i file programmati in ROM disk. Se nella stessa ROM disk è stato salvato un command file con il nome AUTORUN.G80, questo viene automaticamente eseguito secondo le modalità descritte nel paragrafo "PROGRAMMA IN AUTORUN".

FGROM.G80: PROGRAMMAZIONE FLASH EPROM

Per la programmazione della FLASH EPROM è stato realizzato un apposito programma per **GDOS** di nome **FGROM.G80** presente nella directory AUTORUN del disco di lavoro, che in base alle indicazioni dell'utente programma la FLASH EPROM da montata sulla scheda remota. Per l'utilizzo di questo programma devono essere seguite le seguenti fasi:

a) Verificare la configurazione hardware della scheda remota, in porticolare per i jumpers che riguardano la selezione dei dispositivi di memoria, provvedendo ad impostare la configurazione per la FLASH EPROM da programmare.

b) Assicurarsi che tutti i files da porre in ROM disk, siano presenti nelle directory attive dei drive di cui dispone il P.C. o nei drive locali della scheda remota. Infatti il programma **FGROM** gestisce tutti i drive gestiti dallo stesso **GDOS**.

c) Eseguire il programma **FGROM** sulla scheda remota provvedendo a caricarlo da uno dei driver gestiti dal **GDOS**. Al termine di questa fase compare una videata che descrive il programma e ne fornisce le informazioni di versione e release sulla riga di stato in alto. Nell'area di rappresentazione sottostante compare invece un menù con cui l'utente ha la possibilità di selezionare il tipo di scheda remota su cui deve essere programmaata la FLASH EPROM . Infatti il programma **FGROM** si presta alla preparazione delle FLASH EPROM per tutte le schede del carteggio **GRIFO®** provviste dell'apposita circuiteria. La scelta deve essere effettuata premendo il tasto associato alla scheda in uso ed una volta effettuata viene riportata

anche sulla linea di stato superiore.

d) A questo punto compare una seconda videata in cui deve essere scelto il tipo di operazione da effettuare. La programmazione della FLASH EPROM può avvenire in tre modi distinti: la creazione di una nuova ROM disk che implica la cancellazione e quindi la perdita dei precedenti files, l'aggiunta di files alla ROM disk preesistente ed infine la cancellazione dell'ultimo file in ROM disk seguita dall'aggiunta di nuovi files. La scelta deve essere effettuata premendo rispettivamente il tasto **C** per creare una nuova ROM disk, **A** per aggiungere alla preesistente o **D** per cancellare l'ultimo e quindi aggiungere. A scelta effettuata il programma effettua alcune operazioni nei confronti della FLASH EPROM, accompagnate da un apposito messaggio di attesa. Nel caso di pressione del tasto **D** viene richiesta una conferma di cancellazione dell'ultimo file in ROM disk a cui l'utente deve rispondere affermativamente con **Y** (il file viene cancellato) o negativamente con **N** (il file rimane in ROM disk). Al termine di questeoperazioni, l'**FGROM** presenta il modo operativo scelto e lo stato attuale della ROM disk (numero di files e spazio libero) in alto. Se durante l'uso della FLASH EPROM vengono verificate delle incongruenze, viene visualizzato un apposito messaggio ed il programma abortisce, riportandosi al **GDOS**.

e) Terminate le fasi descritte, il programma **FGROM** entra nella fase di vera e propria creazione della ROM disk: il programma entra in un ciclo in cui inizialmente richiede se continuare la programmazione od uscire; uscendo si completa la programmazione della FLASH EPROM e si ritorna al **GDOS**, mentre continuando vengono richiesti in sequenza i nomi dei files da salvare con i rispettivi nomi che dovranno avere in ROM disk. In particolare prima viene richiesto il nome del file presente sui drives gestiti dal **GDOS** e poi il nome che tale file dovrà avere in ROM disk; questa prerogativa è molto utile soprattutto perchè fornisce la possibilità di creare il file di AUTORUN in ROM disk, partendo da un files con nome diverso (di norma quello dell'applicazione). Le modalità di inserzione di questi nomi sono ampiamente descritte dallo stesso programma, che oltretutto fornisce un'indicazione aggiornata del numero di files presenti in ROM disk e dello spazio libero rimasto a disposizione. Quando lo spazio libero (che è in relazione al numero ed alle dimensioni dei files già inseriti) è insufficiente per l'inserimento di un altro file, il programma **FGROM** visualizza un apposito messaggio di stato. Al termine dell'acquisizione e programmazione del file l'**FGROM** genera un richiamo acustico che informa l'utente del completamento dell'operazione.

La modalità di cancellazione dell'ultimo file inserito e successiva aggiunta di nuovi files descritta al punto **d** é particolarmente interessante quando l'utente intende effettuare l'aggiornamento del proprio programma applicativo senza dover modificare gli eventuali altri files presenti sulla ROM disk. In questo caso sarà sufficiente programmare il file con l'applicativo in ultima posizione, quindi cancellarlo e poi aggiungerlo.
Da ricordare che il **GDOS** verifica la presenza dei drive locali e quindi della ROM disk solo dopo il reset od il power on, quindi nel caso di una FLASH EPROM priva di ROM disk in cui viene aggiunta con l'**FGROM**, al fine di vederla e gestirla correttamente é necessario resettare o riaccendere la scheda remota in uso.
Nella FLASH EPROM con l'**FGROM** si può programmare la sola area di ROM disk, non l'intero dispositivo. Per questa ragione lo stesso **FGROM** può essere utilizzato su FLASH EPROM in cui sia già presente il **GDOS** (tipo **FGD xxx** descritto nel capitolo "VERSIONI GDOS"); per ragioni di sicurezza al sistema operativo é riservato il primo settore della FLASH EPROM in modo da salvaguardarlo da ogni eventuale cancellazione.

INTERFACCIE PER I/O DIGITALI

Tutte le schede di controllo del carteggio industriale **Grifo®** dispongono di almeno 16 linee di I/O a livello TTL, disponibili su un connettore standardizzato. Nello stesso carteggio sono inoltre disponibili una vasta serie di interfaccie con lo stesso connettore standardizzato di I/O, che possono essere collegate direttamente alla scheda remota di controllo con un semplice flat. Tra queste interfaccie si possono ricordare schede didattiche,interfaccie di I/O di potenza per il collegamento con i segnali del campo, schede con memoria di massa, interfaccie operatore, ecc. Per alcune di queste schede il GDOS si occupa della loro gestione ad alto livello, in modo che l'utente le possa usare senza preoccuparsi dell'hardware di queste interfaccie, sfruttando addirittura apposite istruzioni del linguaggio di programmazione scelto.
Nella seguente tabella sono riportati i connettori da utilizzare per il collegamento della scheda remota con le interfaccie sucessivamente descritte e del relativo cavo di connessione.

SCHEDA REMOTA	CONNETTORE	CAVO DI CONNESSIONE
GPC° 80F	**CN1**	**FLAT 26-20**
GPC° 81F	**CN3**	**FLAT 20-20**
GPC° 011	**CN5**	**FLAT 20-20**
GPC° 15A	**CN2**	**FLAT 20-20**
GPC° 15R	**CN9**	**FLAT 20-20**
GPC° 153	**CN3**	**FLAT 20-20**
GPC° 183	**CN3**	**FLAT 20-20**

FIGURA 11: CONNETTORI PER INTERFACCIE DI I/O

Naturalmente una sola delle interfaccie sucessivamente descritte può essere collegata e quindi usata con il **GDOS** ed in caso di utilizzo di una delle interfaccie le sedici linee di I/O del connettore riportato in tabella non sono più a disposizione utente. Il **GDOS** non effettua alcuna operazione autonomamente su queste linee, in modo da salvaguardare la sicurezza del sistema quando le linee sono gestite per l'applicazione; solo l'utente può decidere di attivare le gestioni del sistema operativo tramite apposite modalità descritte nei seguenti paragrafi.

STAMPANTE LOCALE

Tra le interfaccie gestite dal **GDOS** si devono ricordare la **IAC 01** e **DEB 01** con cui si può gestire una stampante locale collegata direttamente alla scheda remota. La stampante da utilizzare è una generica stampante con interfaccia di comunicazione parallela di tipo CENTRONICS (stampante da ufficio, da pannello, ecc.) che viene collegata ai due port di I/O digitale riportati sul connettore riportato in figura 11. Tali moduli dispongono di una serie di connettori con pin out standardizzato, quindi dal punto di vista hardware si possono utilizzare dei cavi standard di collegamento con conseguente risparmio economico. Dal punto di vista software, per il comando della stampante possono essere utilizzate tutte le istruzioni che il linguaggio di programmazione mette a disposizione

(LST per il PASCAL, LPRINT per lo ZBASIC, ecc.) con cui si possono direttamente visualizzare stringhe, costanti, variabili ed anche effettuare una rappresentazione grafica. L'utente si deve solo preoccupare di settare opportunamente la struttura dati IOBYTE descritta nell'apposito paragrafo di questo manuale.
Le linee di I/O utilizzate per la gestione della stampante, vengono settate solo durante la stampa del primo carattere che avviene dopo il reset o power on della scheda remota, quindi se non si utilizza la stampante dal punto di vista software il **GDOS** non interviene sulle linee di I/O. La procedura di stampa del **GDOS** non é sospensiva: se la stampante non é pronta, dopo un timeout, la stessa procedura viene abbandonata; non é inoltre possibile avere informazioni sullo stato della stampante.

RAM DISK CON MCI 64

Come già descritto nel paragrafo "RAM ROM DISK" il **GDOS** può gestire una RAM card tramite l'apposita scheda d'interfaccia per memory card PCMCIA: **MCI 64**. Tale drive di RAM disk é associato al drive logico L e può avere dimensioni variabili da un minimo di 512 KByte ad un massimo di 16 MByte. Il collegamento deve essere effettuato seguendo le indicazioni della figura 11 e provvedendo a collegare l'altra estremità del flat collegato alla scheda remota, al connettore CN2 della **MCI 64**.
I LEDs della **MCI 64** indicano rispettivamente:

LD1 di colore rosso	->	segnala una operazione di scrittura sulla memory card (attivo se scrittura in corso e viceversa);
LD2 di colore giallo	->	segnala la presenza della tensione di alimentazione sulla memory card (acceso se tensione presente e viceversa);
LD3 di colore verde	->	segnala lo stato di carica della batteria della memory card (acceso se batteria carica e viceversa);
LD4 di colore rosso	->	segnala lo stato di write protect della memory card (acceso se protezione attiva e viceversa),

e vengono opportunamente settati solo in corrispondenza di un accesso alla RAM card. La memory card può essere disinserita in qualsiasi momento, ammesso che il **GDOS** non vi stia accedendo ed in corrispondenza di questa operazione tutti i LEDs si disattivano. Da ricordare che il LED LD1 si attiva anche in corrispondenza di operazioni di lettura dalla RAM card, infatti se quest'ultima non é protetta in scrittura, viene aggiornata la directory che a sua volta risiede sulla stessa card.
In caso di RAM card con write protect attivo, non possono naturalmente essere effettuate operazioni di scrittura e possono essere mantenuti contemporaneamente aperti solo 10 files sul drive L.
A tutti gli effetti la RAM card gestita dal **GDOS** può essere paragonata ad un normale floppy disk, infatti anche se gestita localmente può essere prelevata e trasportata, può essere utilizzata per effettuare interscambi di files tra diversi sistemi, basati anche su schede remote diverse, é molto capiente e veloce, ecc.
Visto che il **GDOS** verifica la presenza del file AUTORUN.G80 anche sul drive L (prerogativa molto interessante per le schede remote sprovviste di RAM disk di bordo, come la **GPC® 81F**), al fine di non effettuare settaggi indesiderati sulle 16 linee di I/O che gestiscono la **MCI 64**, quando queste sono utilizzate per l'interfacciamento con il campo, si deve informare il sistema operativo **GDOS** della presenza della **MCI 64**. Questa operazione é effettuata provvedendo a salvare un file di configurazione CONFIG.SYS in uno dei drive locali della scheda remota (ROM o RAM disk). Per la creazione di questo file si deve realizzare l'apposito programma per **GDOS**: **CONFIG.G80**, mentre per il suo salvataggio sui drive locali possono essere utilizzate le tecniche descritte nei

precedenti paragrafi. E' quindi l'utente che, tramite questa configurazione, stabilisce se la **MCI 64** è collegata e deve essere quindi gestita; un semplice collegamento dell'interfaccia e della memory card non sono sufficienti affinché questa venga gestita correttamente. Anche il file CONFIG.SYS viene verificato ed utilizzato dal sistema operativo solo in seguito ad un reset o power on della scheda, quindi a seguito della creazione e del salvataggio del file di configurazione si deve far ripartire il **GDOS**.
Il paragrafo "RAM ROM DISK" indica che il **GDOS** quando parte effettua una verifica dei drive locali e per quelli di RAM disk se non li trova formattati, provvede a formattarli autonomamente; questo discorso non é valido per il drive di RAM disk associato alla RAM card, infatti essendo quest'ultima removibile ed utilizzabile anche su sitemi non **GDOS**, si rischierebbe di formattare erroneamente la RAM card utilizzata, con conseguente perdita dei dati precedenti.
Per maggiori informazioni sulle modalità di configurazione del sistema operativo GDOS per la gestione della MCI 64, fare riferimento al paragrafo sucessivo "CONFIG.*: CONFIGURAZIONE DEL GDOS".

INTERFACCIE OPERATORE LOCALI

Il sistema operativo **GDOS** si occupa della gestione di una serie di interfaccie operatori locali che possono essere gestite con le istruzioni ad alto livello del linguaggio di programmazione utilizzato. L'aspetto dell'interfacciamento operatore é sempre uno dei problemi più pesanti in tutti i programmi applicativi, quindi la disponibilità di strumenti già pronti che facilitino questa operazione semplifica il lavoro dell'utente e riduce i tempi di sviluppo. Il dispositivo logico di console primaria gestita dal **GDOS** può essere associata a dispositivi hardware predisposti per l'interfaccia operatore, come **QTP 24P**, **KDx x24**, **IAF 42**, **IAL 42**, **DEB 01**, ecc, che comprendono display alfanumerici, LEDs di stato e tastiera per l'inserimento dati. Tali interfaccie sono provviste di un connettore standardizzato di I/O e può quindi essere direttamente collegata alla scheda remota seguendo le indicazioni della figura 11. Nel caso in cui le interfaccie disponibili nel carteggio **Grifo®** non soddisfino le esigenze dell'utente é possibili realizzare delle proprie interfaccie operatore seguendo le indicazioni dell'appendice A in cui sono riportati gli schemi elettrici di alcune di queste interfaccie.
Per la gestione dei LEDs di stato si deve utilizzare l'apposita procedura di utility descritta nei precedenti paragrafi, mentre per la gestione di tastiera e display si possono utilizzare le istruzioni ad alto livello che gestiscono il dispositivo logico di console, come PRINT, INPUT, INKEY dello ZBASIC; WRITE, READ, KEYPRESSED del PASCAL; PUTS, PRINTF, SCANF, KEYHIT del C; ecc. Prima di utilizzare tali istruzioni, l'utente deve informare il **GDOS** su quale interfaccia é collegata e con quale configurazione, compilando opportunamente l'IOBYTE EXTENSION come di seguito descritto.

IOBYTE EXTENSION: D7 D6 D5 D4 D3 D2 D1 D0

D0 -> Seleziona tipo di display
0 -> Seleziona display LCD
1 -> Seleziona display VFD

Se tipo display = LCD (D0=0):
D1 -> Seleziona tipo di trasmissione
0 -> Seleziona trasmissione ci caratteri al display LCD
1 -> Seleziona trasmissione di comandi al display LCD

Se tipo display = VFD (D0=1):

D1 -> Seleziona tipo d'interfaccia
0 -> Seleziona interfaccia operatore **QTP 24P**
1 -> Seleziona interfaccia operatore **KDx x24**, **IAL 42**, **IAF 42**, **DEB 01**

D2 -> Associa dispositivo logico di console in primaria
0 -> Dispositivo logico di console in primaria associato tramite IOBYTE
1 -> Dispositivo logico di console in primaria associato alla tastiera dell'interfaccia operatore locale

D3 -> Associa dispositivo logico di console out primaria
0 -> Dispositivo logico di console out primaria associato tramite IOBYTE
1 -> Dispositivo logico di console out primaria associato al display dell'interfaccia operatore locale

D4 -> Non utilizzato

D5 -> Non utilizzato

D6 -> Non utilizzato

D7 -> Flag di attivazione handshake per comunicazione con **GET80** (vedere apposito paragrafo)

Con IOBYTE s'intende un'altro parametro del GDOS che agisce sempre sull'associazione dei dispositivi logici di console primaria e viene descritto nel paragrafo "IOBYTE ED IOBYTE EXTENSION".
Con VFD s'intende il tipo di display fluorescente (20x1, 20x2, 20x4, 40x1, 40x2) , mentre con LCD s'intende il tipo di display LCD (20x2, 20x4, 40x1, 40x2) che può essere installato sulle interfaccie elencate. Il **GDOS** gestisce solo i driver firmware per la trasmissione di codici e/o comandi al display scelto ma non si preoccupa di effettuare operazioni di inizializzazione, di gestione del cursore e di gestione dei comandi di rappresentazione. Quest'ultimi devono essere effettuati dal programma pllicativo dell'utente che li può trovare descritti nella documentazione tecnica della casa costruttrice del display. Il **GDOS** si preoccupa autonomamente del settaggio delle linee di I/O per la gestione del display in corrispondenza della prima chiamata alla funzione di console out primaria con D2=1.
La tastiera dell'interfaccia operatore locale é una tastiera a matrice 6x4 per un totale di 24 tasti. Solo le schede remote provviste di counter timer hanno la possibilità di associare il dispositivo logico di console in primaria alla tastiera dell'interfaccia operatore locale e quindi di settare il BIT D3 dell'IOBYTE EXTENSION. Per le altre schede che non hanno questa possibilità (**GPC® 81F**) il BIT D3 é indifferente e l'associazione del dispositivo logico di console in primaria é sempre definita dall'IOBYTE. Per la gestione della tastiera, oltre al timer counter 0, viene attivata anche la gestione in interrupt generato appunto dal timer su una base temporale di 5 msec. Il **GDOS** si preoccupa autonomamente del settaggio di queste condizioni e delle linee di I/O utilizzate, in corrispondenza della prima chiamata alla funzione di console in primaria con D3=1. L'utente si deve limitare a:
- non usare il timer 0;
- nel caso d'uso d'interrupt nel suo applicativo, ad usare il modo vettorizzato ponendo i vettori delle proprie procedure di risposta nell'ultima pagina da 256 Bytes dei 64 KBytes di lavoro. Infatti il **GDOS** carica il valore 0FFH nel registro I e questo non può essere variato;
- non utilizzare i due byte dei 64 KBytes di lavoro da FFF8H÷FFF9H in cui viene salvato il vettore d'interrupt per la gestione della tastiera a matrice.

Al fine di soddisfare tutte le esigenze degli utenti, per la tastiera a matrice dell'interfaccia operatore locale il **GDOS** fornisce la possibilità di definire i codici di codifica dei tasti. Infatti in memoria esiste una tabella di 24 byte in cui possono essere salvati questi codici, come di seguito descritto:

N. TASTO	INDIRIZZO CODICE TASTO	CODICE DEFAULT
0	FF67H	37H = 55 = "7"
1	FF68H	1BH = 27 = "ESC"
2	FF69H	35H = 53 = "5"
3	FF6AH	31H = 49 = "1"
4	FF6BH	0DH = 13 = "CR"
5	FF6CH	30H = 48 = "0"
6	FF6DH	39H = 57 = "9"
7	FF6EH	38H = 56 = "8"
8	FF6FH	36H = 54 = "6"
9	FF70H	34H = 52 = "4"
10	FF71H	33H = 51 = "3"
11	FF72H	32H = 50 = "2"
12	FF73H	4CH = 76 = "L"
13	FF74H	4BH = 75 = "K"
14	FF75H	4AH = 74 = "J"
15	FF76H	49H = 73 = "I"
16	FF77H	48H = 72 = "H"
17	FF78H	47H = 71 = "G"
18	FF79H	46H = 70 = "F"
19	FF7AH	45H = 69 = "E"
20	FF7BH	44H = 68 = "D"
21	FF7CH	43H = 67 = "C"
22	FF7DH	42H = 66 = "B"
23	FF7EH	41H = 65 = "A"

FIGURA 12: INDIRIZZI E VALORI CODICI TASTI TASTIERA A MATRICE

I codici di default che il **GDOS** scrive nella tabella, coincidono con i codici dei tasti serigrafati sulla tastiera della **QTP 24P**.
Per quanto riguarda la corrispondenza tra i numeri tasto riportati nella tabella di figura 12 ed i tasti fisici della tastiera a matrice collegata alle interfaccie operatori locali, vale la corrispondenza riportata nelle figure 13, 14, 15.

PIN CN3 QTP 24P	6	5	4	3	4	1
10	0	4	8	12	16	20
9	1	5	9	13	17	21
8	2	6	10	14	18	22
7	3	7	11	15	19	23

FIGURA 13: NUMERAZIONE TASTI SU QTP 24P

PIN CN2 KDx x24	8	7	6	5	9	10
4	0	4	8	12	16	20
3	1	5	9	13	17	21
2	2	6	10	14	18	22
1	3	7	11	15	19	23

FIGURA 14: NUMERAZIONE TASTI SU KDX X24

CONFIG.*: CONFIGURAZIONE DEL GDOS

Al fine di semplificare le operazioni di settaggio dell'hardware della scheda remota, il **GDOS** fornisce all'utente la possibilità di configurare quest'ultima in modo comodo ed evoluto, grazie ad un apposito file di configurazione. Il **GDOS** in fase di lancio (dopo un reset o power on) verifica la presenza del file di configurazione **CONFIG.SYS** su tutti i drive locali a bordo della scheda remota, con il seguente ordine: N, M, O. Se tale file non viene trovato, provvede ad impostare i settaggi di default che sono:

MCI 64: Non gestita

LINEA SERIALE A		LINEA SERIALE B	
BAUD RATE:	massimo disponibile	BAUD RATE:	19200
BIT x CHR:	8	BIT x CHR:	8
STOP BIT:	2	STOP BIT:	2
PARITY:	NO	PARITY:	NO
HANDSHAKE:	NO	HANDSHAKE:	NO

mentre se il file é presente, viene caricato ed utilizzato per settare le caratteristiche sopra elencate. Come si può notare la configurazione ad alto livello riguarda solo la **MCI 64** e le due linee seriali di comunicazione, infatti queste sono le sezioni hardware comuni a tutte le schede remote che richiedono una procedura di inizializzazione abbastanza articolata. Con handshake s'intende la gestione automatica degli handshake hardware di comunicazione (/CTS ed /RTS), mentre con "massimo disponibile" s'intende il massimo baud rate selezionabile sulla scheda remota in uso.

Questo valore può essere individuato sul manuale tecnico della stessa scheda e deve essere utilizzato in fase di settaggio della velocità di comunicazione sul programma **GET80**.
Per la generazione del file di configurazione si deve utilizzare l'apposito programma per **GDOS** chiamato **CONFIG.G80**, presente nella directory principale del disco di lavoro, che deve essere utilizzato come di seguito descritto.

a) Lanciare il programma CONFIG.G80 dal sistema operativo **GDOS**, provvedendo a caricarlo dal drive su cui é salvato. Al fine della fase di lancio compare una videata di presentazione del programma in cui é riportato anche in numero di versione.

b) Selezionare il tipo di scheda remota su cui si stà lavorando. Per la selezione é sufficiente premere i tasti "freccia destra e sinistra" con cui viene automaticamente variata la scheda scelta e confermare con "ENTER".

c) Selezionare il drive logico su cui s'intende salvare il file di configurazione CONFIG.SYS in creazione. Per la selezione é sufficiente premere i tasti "freccia destra e sinistra" con cui viene automaticamente variato il nome del drive valido per la scheda precedentemente scelta, e confermare con "ENTER".

d) A questo punto compare una videata con i dati presentati nella descrizione dei settaggi di default, in cui l'utente può:
- spostare la voce attualmente scelta (che é rappresentata in modo evidenziato) con i tasti "freccia sù e giù";
- variare il valore della voce attualmente scelta con i tasti "freccia destra e sinistra", con cui i possibili valori per quella voce sono variati e rappresentati in sequenza;
- uscire dal programma di configurazione provvedendo a salvare la configurazione impostata nel file CONFIG.SYS sul drive scelto al punto **c**. Questa operazione é effettuata selezionando la voce "SAVE and EXIT" e confermando con "ENTER" ed é seguita dalla cancellazione della videata e dalla presentazione dell'esito dell'operazione di salvataggio.
- uscire dal programma di configurazione senza salvare la configurazione impostata. Questa operazione é effettuata selezionando la voce "EXIT without SAVE" e confermando con "ENTER" ed é seguita dalla cancellazione della videata.

e) Se al punto precedente é stata scelta l'uscita con salvataggio, sul drive scelto al punto **c** sarà presente il file CONFIG.SYS e se tale file viene copiato su uno dei drive locali (o vi é stato scritto direttamente dal CONFIG.G80) e quindi resettata la scheda remota, i dati di configurazione settati al punto **d** vengono settati sull'hardware.

Le opzioni HANDSHAKE del programma di configurazione, si riferiscono alla gestione automatica degli handshake hardware /CTS, /RTS della scheda remota, durante le fasi di comunicazione (si veda ad esempio la modalità autoenable del dispositivo che gestisce la comunicazione seriale). Queste opzioni comunque non agiscono sul flag di attivazione handshake per la comunicazione con il **GET80**, che quindi rimane a completa gestione utente.
Il file di configurazione CONFIG.G80 viene fornito solo nella versione **xGDOS xxx MCI**, descritta nel capitolo "VERSIONI GDOS", ma può essere esplicitamente richiesto alla **Grifo®**.

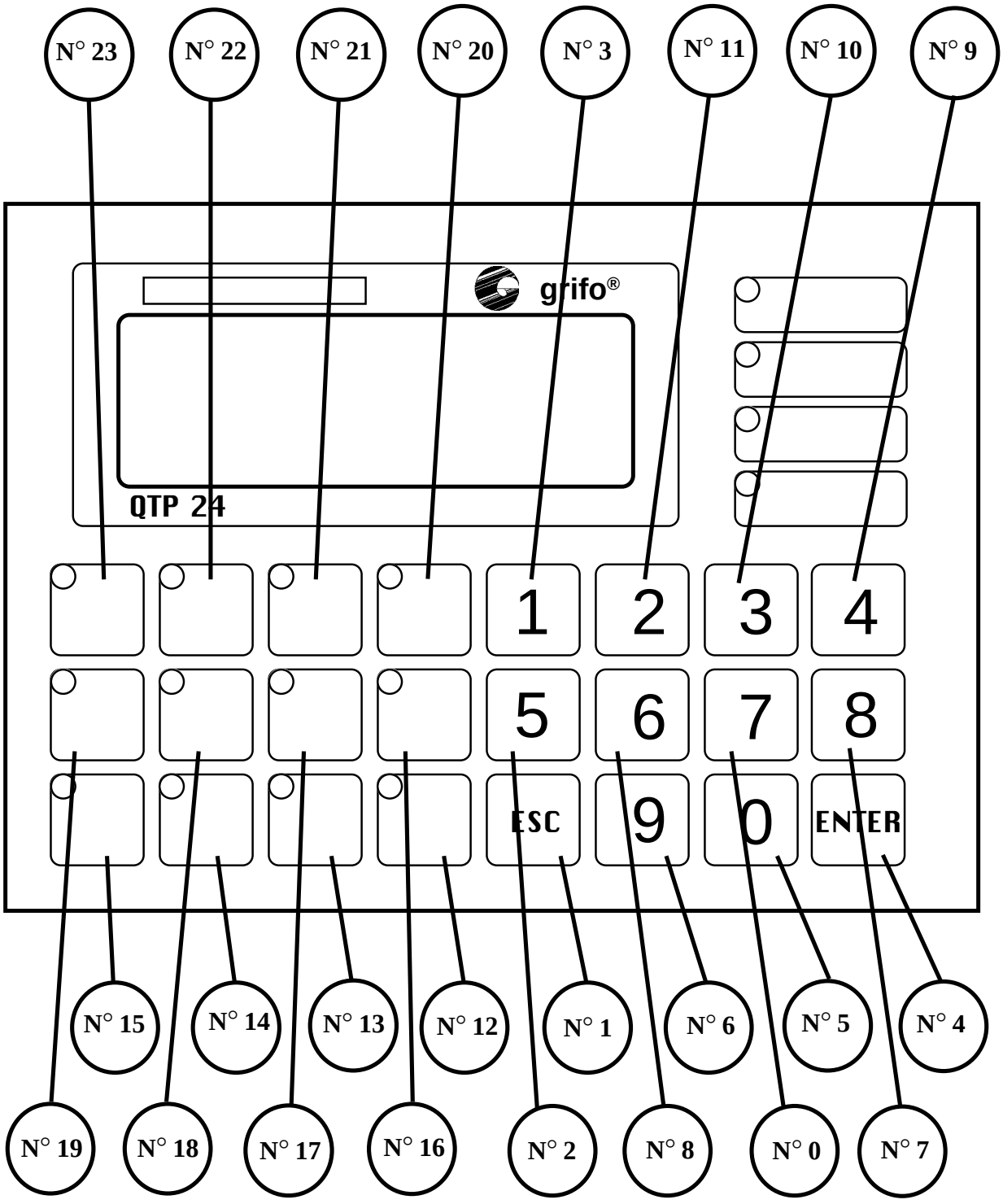


FIGURA 15: DISPOSIZIONE TASTI SU QTP 24P

CARATTERISTICHE TECNICHE DEL GDOS

Vengono di seguito riportate tutte le caratteristiche tecniche del pacchetto software **GDOS**; tali informazioni vengono approfondite nel resto del manuale, di cui questa pagina costituisce un riassunto:

1) Memorie di massa: drive del P.C. da **A** fino ad **C**
drive **L** RAM disk su memory card
drive **M** RAM disk di bordo
drive **N** ROM disk di bordo (su EPROM o FLASH EPROM)
drive **O** RAM Disk di bordo

Tutti questi drive possono avere formati variabili ed in caso della RAM disk viene gestita anche l'eventuale circuiteria di protezione in scrittura.

2) Comunicazione: Linea di console primaria -> Linea seriale primaria

Baud Rate	= massimo settabile, a seconda del tipo di scheda remota
Parità	= Nessuna
Stop Bit	= 2
Bit x Chr.	= 8
Handshake	= /CTS (disattivabile con IOBYTE EXTENSION.7)

Linea di console ausiliaria -> Linea seriale ausiliaria

Baud Rate	= 19200 Baud
Parità	= Nessuna
Stop Bit	= 2
Bit x Chr.	= 8
Handshake	= Nessuno

3) RAM di lavoro: 64 KByte

4) Stampante: Parallela del P.C.
Parallela collegata alla scheda remota

5) Watch Dog: Retriggerato in fase di lettura files

6) EEPROM: Gestita ad alto livello tramite procedure di utility

7) Interfaccie operatore: Display LCD o VFD
Tastiera a matrice
LEDs di stato

VERSIONI GDOS

Come ogni software e firmware, anche il **GDOS** é soggetto a continue evoluzioni e modifiche, con l'intento di soddisfare nel modo migliore le nuove richieste dell'utenza e di eliminare gli eventuali problemi riscontrati. Di base sono disponibili tre diverse implementazioni di **GDOS**:

GDOS xxx
Sistema operativo **GDOS** per la scheda **GPC®** xxx in EPROM.

FGDOS xxx
Sistema operativo **GDOS** per la scheda **GPC®** xxx in FLASH EPROM.

GDOS xxx MCI e **FGDOS xxx MCI**
Sistema operativo **GDOS** per la scheda **GPC®** xxx in EPROM o FLASH EPROM, con gestione del drive di RAM disk su memory card, tramite **MCI 64**.

FGD xxx e **FGD xxx MCI**
FLASH EPROM per la scheda **GPC®** xxx con il sistema operativo **GDOS** programmato nel primo settore.

Per quanto riguarda il numero di versione del GDOS, vale invece la seguente corrispondenza:

Ver. 1.0 -> Prima versione.
Ver. ≥ 1.9 -> Aggiunta gestione drive memoria di massa locali (ROM disk, RAM disk).
Ver. ≥ 2.5 -> Aggiunta gestione EEPROM seriale e stampante locale.
Ver. ≥ 2.6 -> Aggiunta gestione FLASH EPROM.
Ver. ≥ 3.0 -> Aggiunta gestione drive RAM disk su RAM card tramite **MCI 64**.
Ver. ≥ 4.1 -> Modificato protocollo di comunicazione per velocizzare la fase di debug.
Ver. ≥ 4.5 -> Aggiunta gestione interfaccia operatore locale

Ogni eventuale aggiunta o miglioria che l'utente ritiene interessante può essere proposta contattando direttamente la **Grifo®**.

STRUTTURE DATI UTILIZZATE DAL GDOS

Viene di seguito riportata una descrizione delle strutture dati implementate nel **GDOS** e che facilitano la programmazione e quindi l'utilizzo di tutto il pacchetto. Da notare che le strutture di seguito descritte vengono utilizzate per precedenti e sucessivi riferimenti nei vari capitoli e paragrafi del manuale.

F.C.B. (FILE CONTROL BLOCK)

Con tale indicazione si indica una sequenza di 36 byte con cui si individua univocamente un file all'interno di tutte le memorie di massa gestite dal **GDOS**. In particolare il significato di tali byte è il seguente:

0	1	2	3	4	5	6	7	8	9	10	11
DR	F1	F2	F3	F4	F5	F6	F7	F8	T1	T2	T3

12	13	14	15	16	17	18	19	20	21	22	23
EX	S1	S2	RC	D0	D1	D2	D3	D4	D5	D6	D7

24	25	26	27	28	29	30	31	32	33	34	35
D8	D9	DA	DB	DC	DD	DE	DF	CR	R0	R1	R2

FIGURA 16: F.B.C.

dove:

DR = Codice del drive, ovvero il numero del drive a cui si riferisce il file da utilizzare. Tale valore può variare tra 0 (drive di default) e 16 (drive P), infatti il **GDOS** può gestire separatamente fino a 16 drive.

F1÷F8 = Nome del file da utilizzare: corrisponde ai codici ASCII in maiuscolo del nome del file.

T1÷T3 = Estensione del file da utilizzare: corrisponde ai codici ASCII in maiuscolo dell'estensione del file.

EX = Riservato al **GDOS**.

S1 = Riservato al **GDOS**.

S2 = Riservato al **GDOS**.

RC = Riservato al **GDOS**.

D0÷DF = Riservati al **GDOS**.

CR = Riservato al **GDOS**.

R0-R2 = Puntatore al record corrente per accessi random; in effetti tale puntatore è costituito solo dai due byte **R0** e **R1**, infatti **R2** costituisce solo un byte di overflow.

T.P.A. (TRANSIENT PROGRAM AREA)

Area di memoria RAM che è a disposizione dell'utente per i programmi da lui realizzati.In tale area potranno quindi risiedere tutte le informazioni (dati e codice) che l'utente decide di memorizzare. L'organizzazione dei 64K Ram gestiti dalla scheda remota in esecuzione del **GDOS** viene di seguito riportata:

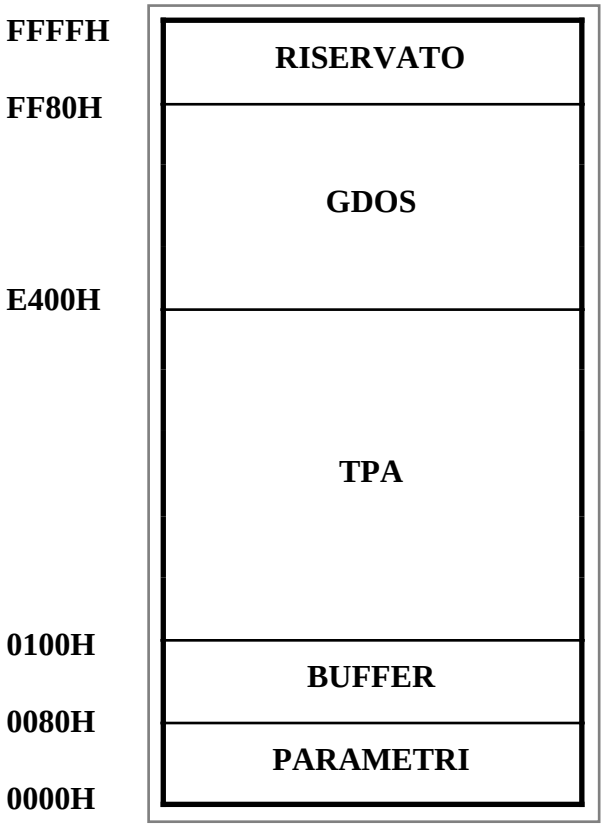


FIGURA 17: MAPPAGGIO MEMORIA

L'area T.P.A. é l'area maggiore dei 64 KBytes di lavoro ed é quella a disposizione dei programmi applicativi del **GDOS**. Quando si utilizzano i linguaggi di programmazione, questi vengono caricati dal **GDOS** nella T.P.A. e quì eseguiti; anche il programma sviluppato dall'utente risiede in quest'area, sia durante lo sviluppo che l'esecuzione finale.
Al fine di disporre sempre di una sezione di RAM non utilizzata dal **GDOS**, è stata implementata una gestione della RAM di lavoro che lascia a disposizione dell'utente un piccolo segmento della stessa. Questa zona, definita **RISERVATO** nella figura sovrastante, può essere modificata solo dall'uso della tastiera dell'interfaccia operatore locale, come indicato nel paragrafo "INTERFACCIA OPERATORE LOCALE", ma normalmente questa é l'unica area che non viene modificata né dal **GDOS** né dai linguaggi di programmazione e può quindi essere usata dall'utente per il salvataggio di dati, il passaggio di parametri tra programmi e procedure, prelevamenti di dati di configurazione, ecc.

I/O BYTE ED IOBYTE EXTENSION

Il **GDOS** dispone di una serie di funzioni che si occupano della gestione ad alto livello di tre dispositivi logici: una console primaria, una console ausiliaria ed una stampante. I linguaggi di programmazione ad alto livellousati dall'utente usano queste funzioni tramite istruzioni dedicate, in modo del tutto trasparente, che facilitano la realizzazione del programma applicativo che ne fà uso. Le funzioni che gestiscono i dispositivi logici sono numerose e vengono descritte dettagliatamente nel capitolo sucessivo, ma possono essere così riassunte:

console primaria -> funzione di input sospensiva, funzione di stato dell'input, funzione di output;
console ausiliaria -> funzione di input sospensiva, funzione di output;
stampante -> funzione di output.

Con le strutture dati IOBYTE ed IOBYTE EXTENSION é possibile associare i dispositivi logici del **GDOS** ad una serie di dispositivi fisici gestiti sempre dal sistema operativo. L'elenco di questi ultimi dispositivi e delle possibili funzioni di gestione é sotto riportata:

linea seriale primaria	->	funzione di input sospensiva, funzione di stato dell'input, funzione di output;
linea seriale ausiliaria	->	funzione di input sospensiva, funzione di stato dell'input, funzione di output;
interfaccia operatore locale	->	funzione di input sospensiva, funzione di stato dell'input, funzione di output;
stampante locale	->	funzione di output;
stampante del P.C.	->	funzione di output.

Con l'uso contemporaneo dell'IOBYTE ed IOBYTE EXTENSION ogni dispositivo logico può essere associato ad ognuno dei dispositivi fisici, con svariate combinazioni e possibilità. In particolare:

I/O BYTE: D7 D6 D5 D4 D3 D2 D1 D0

D0 -> Associa dispositivo logico di console in primaria
0 -> Dispositivo logico di console in primaria associato alla linea seriale primaria
1 -> Dispositivo logico di console in primaria associato alla linea seriale ausiliaria

D1 -> Associa dispositivo logico di console in ausiliaria
0 -> Dispositivo logico di console in ausiliaria associato alla linea seriale ausiliaria
1 -> Dispositivo logico di console in ausiliaria associato alla linea seriale primaria

D3 D2-> Associa dispositivo logico di console out primaria
0 0-> Dispositivo logico di console out primaria associato alla linea seriale primaria
0 1-> Dispositivo logico di console out primaria associato alla linea deriale ausiliaria
1 0-> Dispositivo logico di console out primaria associato alla stampante del P.C.
1 1-> Dispositivo logico di console out primaria associato alla stampante locale

D5 D4-> Associa dispositivo logico di console out ausiliaria
0 0-> Dispositivo logico di console out ausiliaria associato alla linea seriale ausiliaria
0 1-> Dispositivo logico di console out ausiliaria associato alla linea seriale primaria

1 0 -> Dispositivo logico di console out ausiliaria associato alla stampante del P.C.
1 1 -> Dispositivo logico di console out ausiliaria associato alla stampante locale

D7 D6-> Associa dispositivo logico di stampante out
0 0 -> Dispositivo logico di stampante out associato alla linea seriale primaria
0 1 -> Dispositivo logico di stampante out associato alla linea seriale ausiliaria
1 0 -> Dispositivo logico di stampante out associato alla stampante del P.C.
1 1 -> Dispositivo logico di stampante out associato alla stampante locale

IOBYTE EXTENSION: D7 D6 D5 D4 D3 D2 D1 D0

D0 -> Seleziona tipo di display
0 -> Seleziona display LCD
1 -> Seleziona display VFD

Se tipo display = LCD (D0=0):
D1 -> Seleziona tipo di trasmissione
0 -> Seleziona trasmissione ci caratteri al display LCD
1 -> Seleziona trasmissione di comandi al display LCD
Se tipo display = VFD (D0=1):
D1 -> Seleziona tipo d'interfaccia
0 -> Seleziona interfaccia operatore **QTP 24P**
1 -> Seleziona interfaccia operatore **KDx x24**, **IAL 42**, **IAF 42**, **DEB 01**

D2 -> Associa dispositivo logico di console in primaria
0 -> Dispositivo logico di console in primaria associato tramite IOBYTE
1 -> Dispositivo logico di console in primaria associato alla tastiera dell'interfaccia operatore locale

D3 -> Associa dispositivo logico di console out primaria
0 -> Dispositivo logico di console out primaria associato tramite IOBYTE
1 -> Dispositivo logico di console out primaria associato al display dell'interfaccia operatore locale

D4 -> Non utilizzato

D5 -> Non utilizzato

D6 -> Non utilizzato

D7 -> Flag di attivazione handshake per comunicazione con **GET80** (vedere apposito paragrafo)

Un dispositivo fisico può essere associato anche a più dispositivi logici, senza alcuna controindicazione e la variazione delle associazioni (programmazione IOBYTE ed IOBYTE EXTENSION) può avvenire in qualsiasi momento con un risultato immediato. Le associazioni effettuate dall'IOBYTE EXTENSION hanno priorità superiore a quelle effettuate con l'IOBYTE, perciò se il dispositivo logico di console primaria é associato all'interfaccia operatore locale, allora i BIT D0, D2, D3 dell'IOBYTE non sono gestiti dal **GDOS** e quindi indifferenti.

I BIT D7, D1, D0 dell'IOBYTE EXTENSION non riguardano direttamente l'associazione tra dispositivi logici e fisici, ma comunque riguardano indirettamente l'uso del dispositivo logico di console primaria; per maggiori informazioni si vedano i paragrafi "EMULAZIONE TERMINALE" ed "INTERFACCIA OPERATORE LOCALE".
Sia l'IOBYTE che l'IOBYTE EXTENSION sono allocati nell'area di memoria dedicata ai parametri del **GDOS** e rispettivamente agli indirizzi **0003H** e **0008H**. Entrambi i byte possono essere letti e scritti dall'utente sfruttando le istruzioni ad alto livello del linguaggio di programmazione utilizzato.
L'IOBYTE e l'IOBYTE EXTENSION vengono settati dal **GDOS** rispettivamente con i valori 80H e 0F0H con cui si associa il dispositivo logico di console primaria alla linea seriale primaria (sia input che output), il dispositivo logico di console ausiliaria alla linea seriale ausiliaria (sia input che output) ed il dispositivo logico di stampante alla stampante del P.C.
Se ad esempio si desidera effettuare la seguente riassociazione: input da console primaria su linea seriale primaria, output su console primaria su linea seriale ausiliaria, input da console ausiliaria da linea seriale ausiliaria, output su console ausiliaria su linea seriale primaria, output su stampante su stampante locale, provvedere a settare quanto segue:

IOBYTE = 11010100 =D4H = 212
IOBYTE EXTENSION = 10000000 = 80H = 128

Se invece si desidera effettuare la seguente riassociazione: input da console primaria su interfaccia operatore locale, output su console primaria su interfaccia operatore locale, input da console ausiliaria da linea seriale ausiliaria, output su console ausiliaria su linea seriale ausiliaria, output su stampante su stampante locale con un'interfaccia operatore tipo **QTP 24P** con display VFD, provvedere a settare quanto segue:

IOBYTE = 1100xx0x = C0H = 192
IOBYTE EXTENSION = 10001101 = 8DH = 141

RIDIREZIONE DRIVE

Al fine di poter gestire i drive M, N ed O anche con i linguaggi di programmazione che non gestiscono direttamente tutti i drive da 01 a 16 (A - P), è stato implementata a livello del sistema opeartivo **GDOS** la possibilità di ridirezionare i primi drive da A a D. In particolare il byte denominato **DEFDRI** allocato all'indirizzo 0004H indica al sistema operativo il modo da utilizzare per la numerazione dei drive. In particolare:

DRIVE LOGICO		DRIVE FISICO	RIDIREZIONE
DEFDRI = 00H			
A = 01	——————>	A = 01	disattiva
B = 02	——————>	B = 02	disattiva
C = 03	——————>	C = 03	disattiva
:	:	:	:
:	:	:	:
:	:	:	:
O = 15	——————>	O = 15	disattiva
P = 16	——————>	P = 16	disattiva

DRIVE LOGICO	DRIVE FISICO	RIDIREZIONE

DEFDRI = FFH

A = 01	————>	M = 13	attiva
B = 02	————>	N = 14	attiva
C = 03	————>	O = 15	attiva
D = 04	————>	P = 16	attiva

Dove con drive logico si intende il drive utilizzato dall'utente, mentre con drive fisico s'intende il supporto di memoria di massa utilizzato. Il **GDOS** inizializza il byte di ridirezione drive a 00H, ovvero disattivando tutte le ridirezioni.
Quindi se ad esempio si utilizza il basic interpretato **NSB8** che gestisce direttamente solo i drive logici numero 01,02 e 03 e si ha la necessità di utilizzare la Ram e la Rom Disk della scheda remota, è sufficiente:

- settare il byte DEFDRI prima di ogni operazione di utilizzo della Ram Disk: FILL 04,255<CR>
- utilizzare la RAM Disk che è stata associata al drive logico 01: SAVE PROVA.B,1<CR>
- utilizzare la ROM Disk che è stata associata al drive logico 02: LOAD APPL.B,2<CR>

Prima di riportarsi sotto il controllo del **GDOS** ci si deve assicurare di restituire il byte DEFDRI resettato:

FILL 04,00<CR>

FUNZIONI DEL GDOS

Il **GDOS** è provvisto di una serie di funzioni di base con cui ci si interfaccia direttamente all'hardware del sistema senza doverne conoscere le caratteristiche. Per questo sono state realizzate una serie di procedure di base direttamente utilizzabili durante la programmazione. Ogni procedura ha propri parametri d'ingresso e d'uscita, ma tutte le procedure sono accomunate dal modo in cui vengono richiamate:

1- Inizializzazione parametri
2- Selezione funzione tramite registro C dello Z80
3- Chiamata all'indirizzo 5: CALL 0005H
4- La procedura ritorna automaticamente al punto di chiamata.

Tutte queste funzioni sono poste nella zona della RAM di lavoro riservata per il **GDOS** e sono quindi sempre presenti e di conseguenza sempre utilizzabili. Inoltre queste funzioni sono quelle utilizzate anche da tutti programmi di utility e dai vari linguaggi di programmazione. L'uso di queste funzioni è di particolare interesse a tutti gli utenti che intendono sviluppare software di gestione in assembly, in quanto non devono sviluppare tutti i driver d'interfacciamento con l'hardware della scheda remota.Tutte le funzioni interne del **GDOS**, sono descritte, nelle pagine seguenti.

FUNZIONE 0: PARTENZA A CALDO

Parametri di ingresso:

Registro C: 00H

Tale funzione, restituisce il controllo al sistema operativo **GDOS** naturalmente al livello CCP (interprete dei comandi).
Il CCP reinizializza il sub-sistema dei dischi, selezionando e rialloggiando il drive attuale.
Questa funzione ha esattamente lo stesso effetto, che si ottiene effettuando un salto alla locazione BOOT = 0000H.

FUNZIONE 1: INPUT DA CONSOLE PRIMARIA

Parametri di ingresso:

Registro C: 01H

Parametri di uscita:

Registro A: Carattere ASCII

Tale funzione attende il successivo carattere letto dal dispositivo logico di console primaria e lo restituisce nel registro A. Del carattere acquisito viene inoltre effettuato eco sempre sul dispositivo di console primaria.
Tale funzione, restituisce il controllo dell'esecuzione al programma chiamante, solo dopo, che un tasto è stato premuto, ovvero é una funzione sospensiva.

FUNZIONE 2: OUTPUT SU CONSOLE PRIMARIA

Parametri di ingresso:

Registro C: 02H
Registro E: Carattere ASCII

Tale funzione, invia il carattere ASCII, il cui codice é salvato nel registro E, al dispositivo logico di console primaria.

FUNZIONE 3: INPUT DA CONSOLE AUSILIARIA

Parametri di ingresso:

Registro C: 03H

Parametri di uscita:

Registro A: Carattere ASCII

Tale funzione, attende un carattere dal dispositivo logico di console ausiliare, e lo salva nel registro A, senza effettuarne eco.
Il controllo, non è restituito, fino a quando il carattere non è stato letto, ovvero la funzione é sospensiva.

FUNZIONE 4: OUTPUT SU CONSOLE AUSILIARIA

Parametri di ingresso:

Registro C: 04H
Registro E: Carattere ASCII

Tale funzione, invia il carattere ASCII, il cui codice é contenuto nel registro E, al dispositivo logico di console ausiliaria.

FUNZIONE 5: OUTPUT SU STAMPANTE

Parametri di ingresso:

Registro C: 05H
Registro E: Carattere ASCII

Tale funzione, spedisce il carattere ASCII, il cui codice é contenuto nel registro E, al dispositivo logico di stampante.

FUNZIONE 6: OPERAZIONI DIRETTE DI I/O SULLA CONSOLE PRIMARIA

Parametri di ingresso:

Registro C: 06H
Registro E: 0FFH (input) o CARATTERE (output)

Parametri di uscita:

Registro A: carattere o stato.

Prima di chiamare la funzione 6, il registro E, contiene il valore esadecimale FF, che denota una richiesta di input dal dispositivo logico di console primaria, od il codice di un carattere ASCII da rappresentare sul dispositivo logico di console primaria.
Se il valore in ingresso è FF, la funzione 6 restituisce A=00 se nessun carattere è pronto, altrimenti A contiene il carattere inserito dal dispositivo logico di console primaria di cui non effettua eco.
Se il valore in E, non è FF, tale funzione, interpreta il contenuto di E, come un carattere ASCII, che viene fornito al dispositivo logico di console primaria.

FUNZIONE 7: PRELEVA I/O BYTE

Parametri di ingresso:

Registro C: 07H

Parametri di uscita:

Registro A: Valore IOBYTE

Tale funzione, restituisce il valore corrente dell'IOBYTE, nel registro A(vedere apposito paragrafo per la descrizione dell'IOBYTE).

FUNZIONE 8: SETTAGGIO I/O BYTE

Parametri di ingresso:

Registro C: 08H
Registro E: Valore IOBYTE

Tale funzione, cambia il valore dell'IOBYTE, con quello contenuto nel registro E.

FUNZIONE 9: OUTPUT DI UNA STRINGA SU CONSOLE PRIMARIA

Parametri di ingresso:

Registro C: 09H
Registro DE: Indirizzo stringa.

Tale funzione, manda i caratteri della stringa memorizzata in memoria a partire dalla locazione puntata da DE, al dispositivo logico di console primaria, fino a quando non si incontra il carattere $, nella stringa.

FUNZIONE 10: INPUT A BUFFER DALLA CONSOLE PRIMARIA

Parametri di ingresso:

Registro C: 0AH
Registro DE: Indirizzo del buffer

Parametri di uscita:

Caratteri della console primaria nel buffer

Tale funzione, inserisce tutti i caratteri acquisiti dal dispositivo logico di console primaria, in un buffer allocato all'indirizzo espresso dal contenuto del registro DE, fino a che non avviene una overflows del buffer (riempimento), o fino a che non viene acquisito un CARRIAGE RETURN.
Il buffer è così strutturato:

DE:	+0	+1	+2	+3	+4	+5	+6	+7	+8	....	+n
	mx	nc	c1	c2	c3	c4	c5	c6	c7	....	??

FIGURA 18: BUFFER DI RICEZIONE DA CONSOLE

dove **mx** è il numero massimo di caratteri, che il buffer, può memorizzare (da 1 a 254) e **nc** rappresenta il numero di caratteri effettivamente letti dalla console primaria. Le successive locazioni, contengono i caratteri inseriti. Le prime due locazioni (mx e nc) sono contatori, che non considerano il CR finale, che infatti, non viene memorizzato, nel buffer e se nc<mx, le posizioni che seguono l'ultimo carattere salvato sono inutilizzate.
Alcune codici di controllo, sono riconosciute, durante l'acquisizione della stringa:

ctl-C ricarica il **GDOS**, rieseguendo l'interprete comandi;
ctl-H (BACK SPACE) elimina l'ultimo carattere inserito
ctl-M (CR) fa terminare la linea di input

Durante l'acquisizione dei caratteri dal dispositivo logico di console primaria ne viene effettuato eco sullo stesso dispositivo.

FUNZIONE 11: LETTURA STATO DELLA CONSOLE PRIMARIA

Parametri di ingresso:

Registro C: 0BH

Parametri di uscita:

Registro A: Stato della console

Tale funzione, effettua un controllo, che permette di verificare se un carattere, è stato digitato dal dispositivo logico di console primaria, oppure no. Se un carattere è stato digitato, nel registro A, viene caricato il valore 0FFH, altrimenti, viene restituito il valore 00H.

FUNZIONE 12: RESTITUZIONE NUMERO DI VERSIONE

Parametri di ingresso:

Registro C: 0CH

Parametri di uscita:

Registro HL: Numero della versione.

Tale funzione, fornisce informazioni, sul numero della versione. Per mantenere la compatibilità con tutti i linguaggi di programmazione, viene sempre restituito: H = 00H ed L = 22H per emulare la versione 2.2 del sistema operativo CP/M.

FUNZIONE 13: RESET SISTEMA DI GESTIONE DISCHI

Parametri di ingresso:

Registro C: 0DH

Questa funzione, permette di riinizializzare, il sistema che gestisce i drive, in modo che tutti i drive vengano settati per la lettura e la scrittura (vedere funzioni 28 e 29) o comunque sia azzerata una eventuale condizione di errore preesistente.

FUNZIONE 14: SELEZIONE DEL DRIVE

Parametri di ingresso:

Registro C: 0EH
Registro E: Drive da selezionare.

Tale funzione, designa il nome del drive dichiarato nel registro E, ad essere usato come drive di

default, per le successive operazioni di accesso ai files. Con E=0, viene selezionato il drive A, con E=1, il drive B e così via, fino a 15, che corrisponde al drive P in un sistema a 16 drive. Il drive, è posto, in uno stato di on-line e tutti gli accessi a file con FCB riferiti al drive 0, automaticamente, fanno riferimento al drive di default corrente selezionato con questa funzione.

FUNZIONE 15: APERTURA DI UN FILE

Parametri di ingresso:

Registro C: 0FH
Registro DE: Indirizzo del FCB

Parametri in uscita:

Registro A: risultato operazione

Tale funzione apre un file presente nella directory del disco passato con l'FCB. Il **GDOS** effettua una scansione, della directory del disco selezionato, per verificare se esiste il file specificato e se il nome file viene trovato, le informazioni della directory, sono copiate nel FCB, in modo che siano possibili, successive operazioni di lettura o di scrittura.
Da notare, che nessuna operazione su file, può essere fatta, senza avere prima aperto il relativo file.
Prima di restituire il controllo al programma chiamante, tale funzione carica in A un codice che definisce se l'operazione ha avuto successo oppure no.
Più specificatamente, se in A viene restituito il valore 0, l'operazione ha avuto successo, altrimenti in A viene caricato 0FFH che indica che il file non è stato trovato.
Da ricordare, che un punto interrogativo (3FH) per le wild card, indica che in quella posizione, è valido qualsiasi carattere e che può essere usato solo sui drive del P.C.

FUNZIONE 16: CHIUSURA DI UN FILE

Parametri di ingresso:

Registro C: 10H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Tale funzione, effettua l'operazione inversa della funzione 15. L'FCB indirizzato da DE deve essere stato preventivamente attivato attraverso una operazione di apertura o di generazione (vedi funzioni 15 o 22) e questa funzione registra permanentemente il nuovo file sul disco specificato. Il processo di confronto dell'FCB della funzione di close, è identico a quello che avviene per la funzione OPEN.
Il risultato dell'operazione restituito da una operazione di chiusura che ha avuto successo é 0, mentre se il nome file, non è presente nella directory, il valore restituito è 0FFH (255 decimale).
Un file non richiede la chiusura, se sono state effettuate solo operazioni di lettura, viceversa si rende necessario registrare permanentemente le nuove informazioni nella directory.

FUNZIONE 17: RICERCA DEL PRIMO FILE

Parametri di ingresso:

Registro C: 11H
Registro DE: indirizzo dell'FCB

Parametri di uscita:

Registro A: risultato operazione

Questa funzione, effettua una scansione della directory, ed effettua un confronto con il file fornito dall'indirizzo del FCB presente in DE. Se il file non viene trovato il valore restituito risulta essere 255, in caso contrario il valore restituito é 0. In quest'ultimo caso, il corrente indirizzo DMA viene compilato con il FCB del file trovato. Sebbene i programmi applicativi normalmente non lo richiedono, le informazioni del FCB possono essere estratte dal buffer DMA dalle relative posizioni. Le wild card, caratteri ASCII "?" (63 = 3FH) o "*" (42 = 2AH), in qualsiasi posizione da f1 ad ex significa che qualsiasi carattere è valido nel confronto e sono gestite solo sui drive del P.C. Anche le wild card abbinate a questa funzione non sono normalmente usate dai programmi applicativi, ma essa ammette una completa flessibilità nella scansione di tutti i file del disco specificato.

FUNZIONE 18: RICERCA DEL SUCCESSIVO FILE

Parametri di ingresso:

Registro C: 12H

Parametri di uscita:

Registro A: risultato operazione

Tale funzione, è simile alla funzione 17, tranne che la scansione continua, dall'ultimo confronto effettuato. Anche questa funzione restituisce il valore decimale 255 in A se la ricerca non ha avuto successo e 0 nel caso contrario.

FUNZIONE 19: CANCELLAZIONE DI UN FILE

Parametri di ingresso:

Registro C: 13H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Tale funzione cancella il file, specificato dal FCB indirizzato da DE. Il nome e il tipo del file, possono contenere wild card, con conseguente canncellazione di tutti i file che rispondono alle specifiche fornite, ma solo sui drive del P.C.
Tale funzione restituisce il valore decimale 255 se non sono stati trovati i file (o il file), altrimenti fornisce il valore 0.

FUNZIONE 20: LETTURA SEQUENZIALE

Parametri di ingresso:

Registro C: 14H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Supponendo che il FCB indirizzato da DE sia stato precedentemente attivato da una operazione di apertura o di generazione (funzione 15 o 22), tale funzione legge i successivi 128 byte del record corrente del file scelto e li trasferisce in memoria, dalla posizione indicata dal corrente indirizzo DMA.
Il record è letto dalla posizione attuale che è automaticamente incrementata al successivo record.
Tale funzione restituisce nel registro A, il valore 00H se l'operazione di lettura ha avuto successo, il valore 01H se é stata raggiunta la fine del file od il valore FFH se la lettura non ha avuto successo.

FUNZIONE 21: SCRITTURA SEQUENZIALE

Parametri di ingresso:

Registro C: 15H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Supponendo che il FCB indirizzato da DE, sia stato attivato attraverso una operazione di apertura o di generazione (funzioni 15 o 22), tale funzione scrive i 128 byte presenti all'indirizzo DMA corrente, sul file specificato dall'FCB, nel record corrente.
Il puntatore al record corrente è successivamente incrementato in modo che punti al record successivo.
L'operazione di scrittura può comportare la perdita delle informazioni precedentemente registrate sul file, infatti la scrittura avviene sul record corrente e per questo bisogna fare attenzione a quale é il suo valore.
In caso che l'operazione di scrittura abbia avuto successo, nel registro A come parametro di ritorno viene inserito 00H, in caso contrario si ha il valore FFH.

FUNZIONE 22: CREAZIONE DI UN FILE

Parametri di ingresso:

Registro C: 16H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Tale funzione è simile a quella di open, eccetto che il FCB deve specificare un nome file non presente sul disco selezionato. Il **GDOS** crea il file, aggiorna la directory ed inizializza il FCB del file in modo che risulti essere vuoto.
Il programmatore deve accertarsi che non esistano altri file aventi lo stesso nome od eventualmente è sufficiente effettuare una operazione di delete del file in questione.
Se l'operazione ha successo tale funzione restituisce nel registro A il valore 0, altrimenti il valore FFH (255 decimale).
Tale funzione effettua come sopra descritto l'attivazione del FCB, perciò per operare successivamente sul file, non è necessario effettuare una operazione di OPEN.

FUNZIONE 23: CAMBIO DEL NOME DI UN FILE

Parametri di ingresso:

Registro C: 17H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Tale funzione, sull'eventuale file specificato dai primi 16 bytes del FBC, sostituisce il nome e l'estensione file con i successivi 16 bytes sempre del FCB passato in ingresso, in modo che il vecchio nome, venga sostituito dal nuovo.
Il codice drive dr, alla posizione 0, è usato per selezionare il drive, mentre il codice del drive per il nuovo nome file alla posizione 16 del FCB non é utilizzato e deve essere 0.
Al ritorno da questa funzione, il registro A contiene il valore 0 se l'operazione ha avuto successo, altrimenti il valore FFH se il nome file non è stato trovato nel disco scelto.

FUNZIONE 24: RESTITUZIONE VETTORE DEI DRIVE ATTIVI

Parametri di ingresso:

Registro C: 18H

Parametri di uscita:

Registro HL: vettore dei drive attivi

Tale funzione fornisce un vettore nel registro HL in cui lo stato dei bit indica quali drive sono attualmente attivi e quindi gestiti dal **GDOS**. Uno stato 1 del bit equivale allo stato di drive attivo del corrispondente drive e viceversa. Il bit meno significativo (L.1) corrisponde al drive A, mentre quello più significativo (H.7) corrisponde al drive P.

FUNZIONE 25: DRIVE CORRENTE

Parametri di ingresso:

Registro C: 19H

Parametri di uscita:

Registro A: numero indicante il drive corrente

Tale funzione restituisce un numero che indica il drive corrente. Il suddetto numero, può variare nell'intervallo 0÷15 al quale corrispondono i relativi drive A÷P.

FUNZIONE 26: SETTAGGIO INDIRIZZO DMA

Parametri di ingresso:

Registro C: 1AH
Registro DE: indirizzo DMA

Il DMA (direct memory address) è un particolare tipo di indirizzamento che viene utilizzato nella gestione di drive di memoria di massa con accesso diretto alla memoria della scheda remota. In **GDOS**, tale indirizzo punta all'area di 128 byte, che verranno sucessivamente scritti oppure letti dal disco. Dopo una partenza a freddo, o dopo un reset, tale indirizzo è settato a 0080H. Questa funzione può essere utilizzata per cambiare questo valore di default, in modo che punti ad un'altra area di memoria specificata dal programma chiamante; in questo modo il programma chiamante nelle operazioni di lettura e/o scrittura di un file non si deve preoccupare di muovere i dati letti o da scrivere in quanto il GDOS già li pone o li prende dove indicato con questa funzione.

FUNZIONE 27: LETTURA INDIRIZZO DEL VETTORE DI ALLOCAZIONE

Parametri di ingresso:

Registro C: 1BH

Parametri di uscita:

Registro HL: indirizzo vettore di allocazione drive

Un vettore di allocazione è mantenuto nella memoria del **GDOS**, per ogni drive attivo (on line). Normalmente, questa funzione, non è utilizzata dai programmi applicativi, ma solo da programmi di sistema o linguaggi di programmazione che fanno uso delle informazioni contenute in questo vettore, per determinare la quantità di spazio libero su disco.
Il **GDOS** restituisce l'indirizzo di un vettore di allocazione fisso che non fornisce indicazioni veritiere nei confronti dei drive gestiti.

FUNZIONE 28: PROTEZIONE IN SCRITTURA DEL DISCO

Parametri di ingresso:

Registro C: 1CH

Tale funzione, effettua una protezione in scrittura temporanea del drive corrente.
Un qualsiasi tentativo di scrittura, prima di una successiva partenza a freddo o di una operazione di reset, comporta la restituzione di un errore in corrispondenza delle operazioni di scrittura sul drive settato di sola lettura.

FUNZIONE 29: LETTURA VETTORE DRIVE DI SOLA LETTURA

Parametri di ingresso:

Registro C: 1DH

Parametri di uscita:

Registro HL: vettore dei drive in read only

Tale funzione fornisce un vettore di bit nel registro HL in cui lo stato indica quali drive sono momentaneamente settati per la sola lettura. Uno stato 1 del bit equivale allo stato di read only del corrispondente drive e viceversa. Come per la funzione 24, il bit meno significativo corrisponde al drive A, mentre quello più significativo corrisponde al drive P.
I bit di read only, vengono settati quando viene effettuata una chiamata esplicita alla funzione 28.

FUNZIONE 30: SETTAGGIO ATTRIBUTI DI UN FILE

Parametri di ingresso:

Registro C: 1EH
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: risultato operazione

Tale funzione permette di agire direttamente sugli attributi dei files. In particolare il read only e gli attributi di sistema (t1 e t2) possono essere settati o resettati. Il FCB indirizzato da DE contiene un nome file non ambiguo e gli appropriati attributi; la funzione 30 effettua una ricerca ed un confronto, fino a che non trova il file nella directory, quindi sostituisce i vecchi attributi, con quelli specificati. Questa funzione ha effetto solo sui files salvati sui drive del P.C., su quelli locali gli attributi vengono settati ma non gestite. In A viene restituito il valore 0 nel caso di file trovato e settaggio attributi effettuato e FFH in tutti gli altri casi.

FUNZIONE 31: PRELEVA INDIRIZZO DEL D.P.B.

Parametri di ingresso:

Register C: 1FH

Parametri di uscita:

Registro HL: indirizzo del DPB

Questa funzione pone in HL, l'indirizzo dei parametri del drive (Disk Parameter Block), residente nel **GDOS**. Tale indirizzo può essere utilizzato per due scopi: primo, i valori dei parametri del disco, possono essere estratti e visualizzati, in modo da avere informazioni dirette sullo stato del disco ossia per esempio, quanto spazio utilizzabile rimane; secondo, cambiare i valori dei parametri del disco corrente, in base al cambiamento del disco. Di norma i programmi applicativi non fanno uso di questa funzione.
Il **GDOS** restituisce l'indirizzo di un blocco parametri standard che non fornisce indicazioni veritiere nei confronti dei drive gestiti.

FUNZIONE 32: SETTAGGIO O LETTURA DEL CODICE USER

Parametri di ingresso:

Registro C: 20H
Register E: 0FFH (lettura) o CODICE USER (settaggio)

Parametri di uscita:

Registro A: user corrente o nessun valore

Un programma applicativo può cambiare o interrogare il numero USER correntemente attivo, effettuando una chiamata alla funzione 32. Se il registro E contiene 0FFH, il valore del corrente numero USER viene inserito nel registro A (parametro di uscita compreso nell'intervallo 0÷15), in caso contrario la USER precedentemente attiva viene sostituito da quella presente in E (modulo 16). Il **GDOS** non gestisce le USER ed in caso di lettura restituisce sempre il valore 0.

FUNZIONE 33: LETTURA RANDOM

Parametri di ingresso:

Registro C: 21H
Registro DE: indirizzo dell'FCB

Parametri di uscita:

Registro A: risultato operazione

Tale funzione è simile all'operazione di lettura di un file sequenziale , eccetto che l'operazione di lettura inizia ad un particolare numero di record, definito dai 24 bit salvati nei 3 bytes del FCB (r0 a 33, r1 a 34, e r2 a 35). Da notare che il byte meno significativo è il primo (r0) mentre quello più significativo è l'ultimo (r2). Il byte r2 deve essere a 0, ma un valore diverso da zero indica un overflow, cioè che si è sulla fine file. Da questo si ricava che r0 ed r1 sono trattati come una word, che contiene il record da leggere. Questo valore, può essere compreso da 0 a 65535, in modo che l'accesso a qualsiasi record degli 8 megabyte del file sia possibile.
Al ritorno dalla chiamata, il registro A contiene il codice di errore FFH in caso di file non presente o errori di lettura, il valore 01H in caso di lettura dopo la fine del file od il valore 00 nel caso di operazione eseguita con successo; negli ultimi due casi l'indirizzo DMA corrente, contiene il record letto in modo random.
Da notare, che contrariamente a quello che avviene per la operazione di lettura sequenziale, il numero record, non viene incrementato. Perciò, in caso si effettui una lettura random seguita da una lettura sequenziale, verrà letto sempre lo stesso record. L'utente può naturalmente far avanzare la posizione del record random, tramite operazioni random di lettura e scrittura, emulando così gli effetti delle operazioni sequenziali di accesso, semplicemente incrementando il valore di r0, r1 per ogni chiamata alla funzione.

FUNZIONE 34: SCRITTURA RANDOM

Parametri di ingresso:

Registro C: 22H
Registro DE: indirizzo dell'FCB

Parametri di uscita:

Registro A: risultato operazione

Tale operazione è inizializzata, in modo simile alla funzione 33, a differenza di quest'ultima però, i dati vengono scritti sul file prelevandoli dall'indirizzo DMA corrente. Come per la funzione 33 il numero record, non viene variato dopo la scrittura.
L'utente può naturalmente far avanzare la posizione del record random, tramite operazioni random di lettura e scrittura, emulando così gli effetti delle operazioni sequenziali di accesso, semplicemente incrementando il valore di r0, r1 per ogni chiamata alla funzione.
Il codice di errore restituito da tale funzione, può assumere il valore 0 se la funzione é stata eseguita con successo oppure il valore 06H in caso di riempimento del disco o 0FFH in caso di errore.

FUNZIONE 35: CALCOLO LUNGHEZZA DEL FILE

Parametri di ingresso:

Registro C: 23H
Registro DE: indirizzo FCB

Parametri di uscita:

dimensione del file, in r0,r1 e r2

Tale funzione può essere utilizzata per determinare la lunghezza di un file identificato dal FCB puntato da DE.
Al ritorno dalla funzione, il puntatore al record corrente r0, r1, r2 del FCB viene settato con il numero dell'ultimo record del file. Così considerando i due byte r0, r1 come una unica parola a 16 bit (dove r0, rappresenta i bit meno significativi) e moltiplicandola per la dimensione di un record (128 bytes) si ottiene la lunghezza effettiva del file.
Ulteriori dati, possono essere aggiunti alla fine di un file già esistente: tale operazione la si effettua tramite una chiamata alla funzione 35 e successivamente, eseguendo scritture random a partire dal puntatore record ottenuto. In caso di file non presente sul disco indicato nel FCB vengono restituiti i tre byte r0, r1, r2 azzerati.

FUNZIONE 36: ACQUISIZIONE DEL RECORD RANDOM

Parametri di ingresso:

Registro C: 24H
Registro DE: indirizzo FCB

Parametri di uscita:

puntatore al record attuale in r0, r1 e r2

Tale funzione può essere utilizzata per determinare la posizione del record attuale in un file identificato dal FCB puntato da DE.
Al ritorno dalla funzione, il puntatore al record corrente r0, r1, r2 del FCB viene settato con il numero del record attuale del file. La funzione principle di questa funzione é quella di ottenere l'attuale posizione del record attuale di un file, ad esempio in seguito ad accessi sequenziali e passare quindi ad operazioni con accessi random.
In caso di file non presente sul disco indicato nel FCB vengono restituiti i tre byte r0, r1, r2 azzerati.

FUNZIONE 37: RESET DEL DRIVE

Parametri di ingresso:

Registro C: 25H
Registro DE: vettore drive da resettare

Parametri di uscita:

Registro A: 00H

Tale funzione, effettua un reset, dei drive specificati nel vettore d'ingresso.
Il parametro passato, è un vettore a 16 bit, che indica appunto quali drive devono subire questa operazione (ricordare che il bit, meno significativo, corrisponde al drive A) in cui il bit a 1 corrisponde ad un drive da resettare.

FUNZIONE 40: SCRITTURA RANDOM CON INSERIMETO DI ZERI

Parametri di ingresso:

Registro C: 28H
Registro DE: indirizzo FCB

Parametri di uscita:

Registro A: Risultato operazione

Tale funzione, è simile alla funzione 34, con l'eccezione che il blocco è riempito di zeri, prima della scritura dei dati.

FUNZIONE	INPUT	OUTPUT
Partenza a caldo	C = 00H	-----
Input da console primaria	C = 01H	A = Dato Ricevuto
Output su console primaria	C = 02H E = Dato da trasmettere	-----
Input da console ausiliaria	C = 03H	A = Dato Ricevuto
Output su console ausiliaria	C = 04H E = Dato da trasmettere	-----
Output su stampante	C = 05H E = Dato da trasmettere	------
Operazioni dirette di I/O sulla console primaria	C = 06H E = FFH (input) o E = Dato da trasmettere	A = Dato ricevuto
Preleva IOBYTE	C = 07H	A = IOBYTE
Settaggio IOBYTE	C = 08H E = IOBYTE	-----
Output di una stringa su console primaria	C = 09H DE = Indirizzo stringa	-----
Input a buffer dalla console primaria	C = 0AH DE = Indirizzo buffer	Buffer aggiornato
Lettura stato della console primaria	C = 0BH	A = Stato della console
Restituzione del numero di vcersione	C = 0CH	HL = Numero versione
Reset sistema di gestione dischi	C = 0DH	-----
Selezione del drive	C = 0EH E = Drive da selezionare	-----
Apertura di un file	C = 0FH DE = Indirizzo FCB	A = Risultato operazione
Chiusura di un file	C = 10H DE = Indirizzo FCB	A = Risultato operazione
Ricerca del primo file	C = 11H DE = Indirizzo FCB	A = Risultato operazione
Ricerca del sucessivo file	C = 12H DE = Indirizzo FCB	A = Risultato operazione
Cancellazione di un file	C = 13H DE = Indirizzo FCB	A = Risultato operazione
Lettura sequenziale	C = 14H DE = Indirizzo FCB	A = Risultato operazione
Scrittura sequenziale	C = 15H DE = Indirizzo FCB	A = Risultato operazione
Creazione di un file	C = 16H DE = Indirizzo FCB	A = Risultato operazione

FUNZIONE	INPUT	OUTPUT
Cambio del nome di un file	C = 17H DE = Indirizzo FCB	A = Risultato operazione
Restituzione vettore dei drive attivi	C = 18H	HL = Vettore drive attivi
Drive corrente	C = 19H	A = Drive corrente
Settaggio indirizzo DMA	C = 1AH DE = Indirizzo DMA	-----
Lettura indirizzo del vettore di allocazione	C = 1BH	HL = Vettore allocazione drive
Protezione in scrittura del disco	C = 1CH	------
Lettura vettore drive di sola lettura	C = 1DH	HL = Vettore drive read only
Settaggio degli attributi di un file	C = 1EH DE = Indirizzo FCB	A = Risultato operazione
Preleva indirizzo del D.P.B.	C = 1FH	HL = Indirizzo del D.P.B.
Settaggio o lettura del codice user	C = 20H	A = User corrente
Lettura random	C = 21H DE = Indirizzo FCB	A = Risultato operazione
Scrittura random	C = 22H DE = Indirizzo FCB	A = Risultato operazione
Calcolo lunghezza del file	C = 23H DE = Indirizzo FCB	r0,r1,r2 = Lunghezza file
Acquisizione del record random	C = 24H DE = Indirizzo FCB	r0,r1,r2 = Numero record
Reset del drive	C = 25H DE = Vettore drive	A = 00H
Scrittura random con inserimento di 0	C = 0FH DE = Indirizzo FCB	A = Risultato operazione

FIGURA 19: TABELLA DELLE FUNZIONI GDOS

APPENDICE A: SCHEMI INTERFACCIE OPERATORE LOCALI

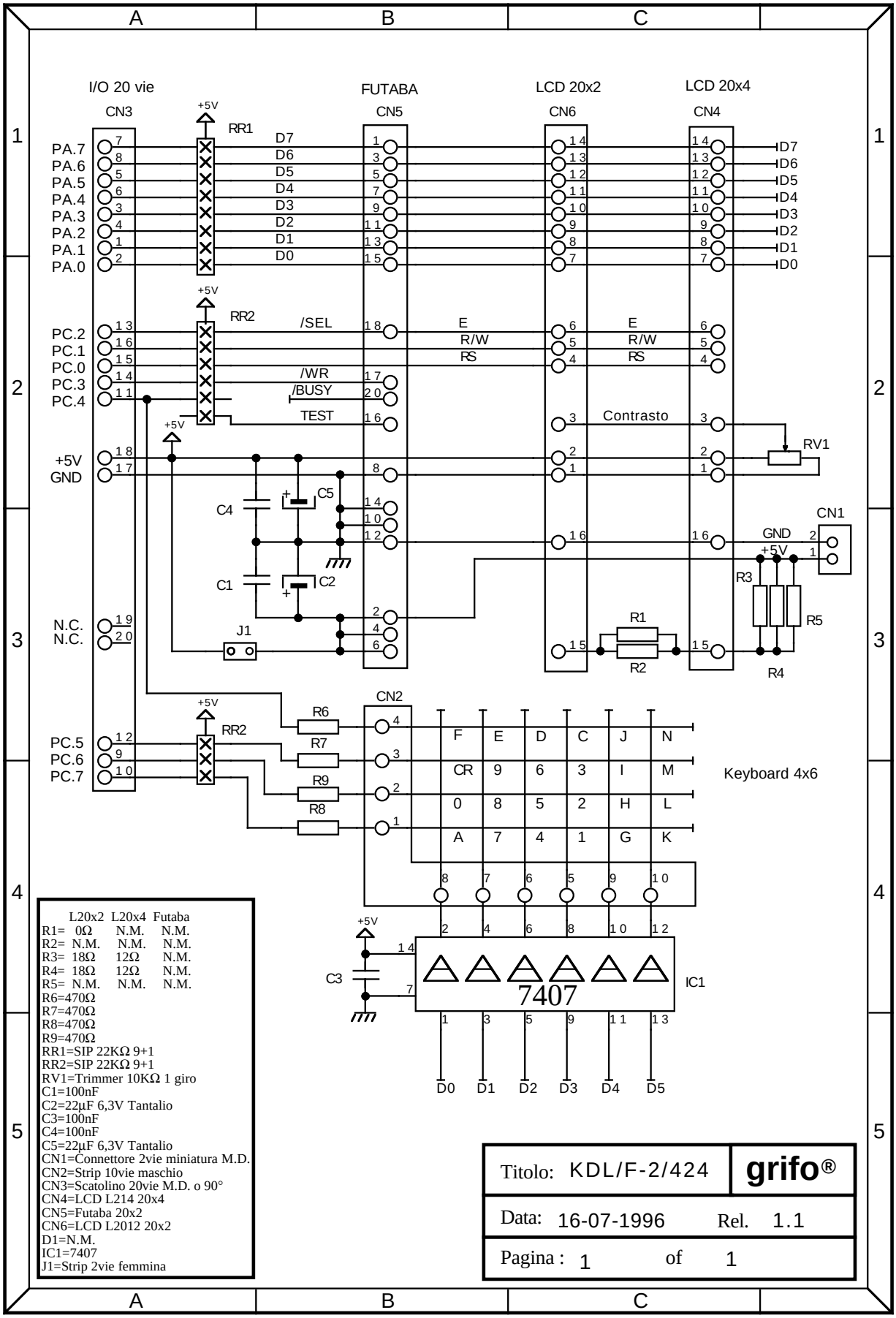
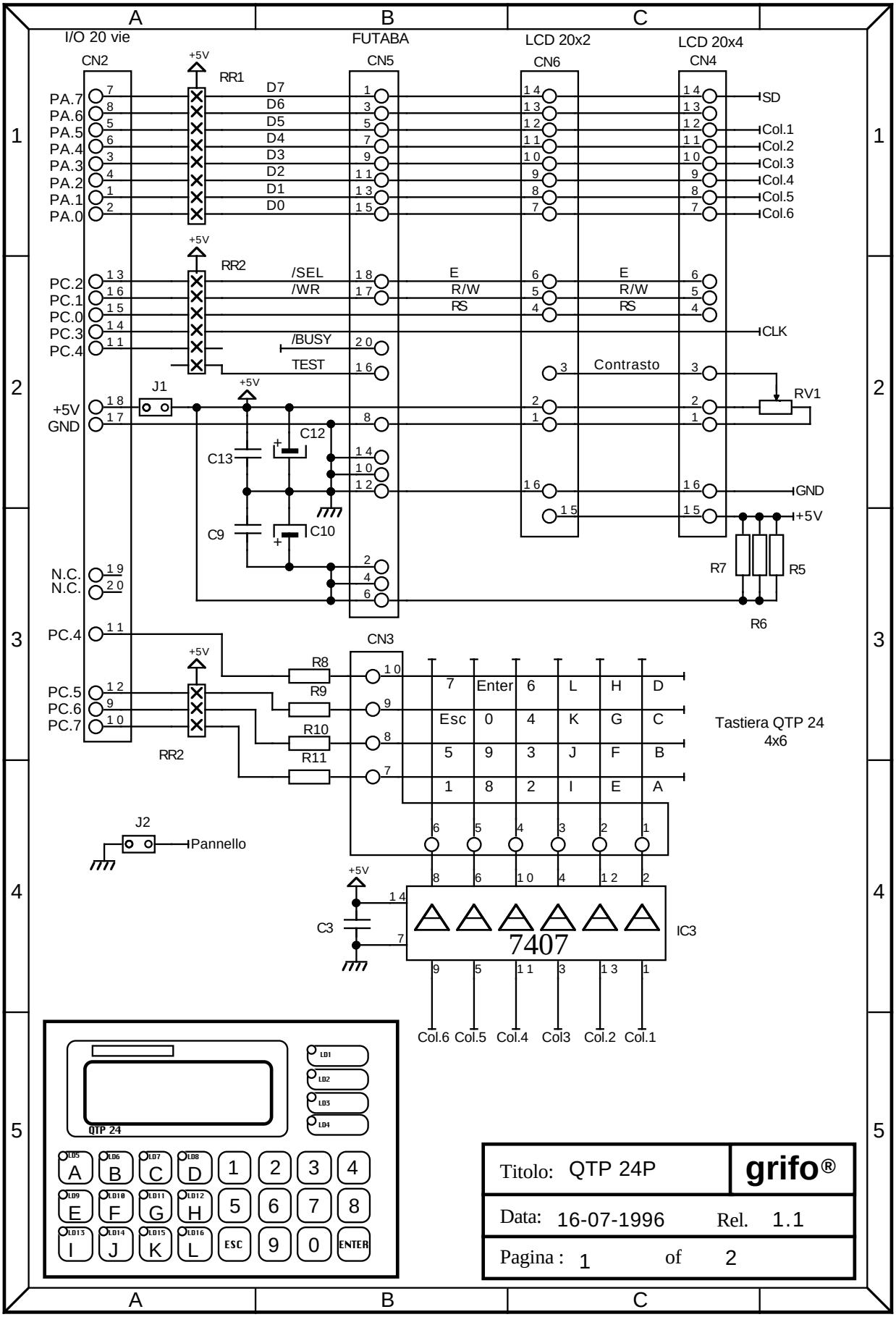


FIGURA 20: SCHEMA ELETTRICO KDX X24


A
B
C
I/O 20 vie
FUTABA
LCD 20x2
LCD 20x4
CN2
CN5
CN6
CN4
+5V
RR1
D7
D6
D5
D4
D3
D2
D1
D0
PA.7
PA.6
PA.5
PA.4
PA.3
PA.2
PA.1
PA.0
SD
Col.1
Col.2
Col.3
Col.4
Col.5
Col.6
RR2
/SEL
/WR
/BUSY
TEST
E
R/W
RS
CLK
PC.2
PC.1
PC.0
PC.3
PC.4
Contrasto
RV1
J1
+5V
GND
C12
C13
C9
C10
GND
+5V
R7
R5
R6
N.C.
N.C.
PC.4
CN3
R8
R9
R10
R11
PC.5
PC.6
PC.7
RR2
Tastiera QTP 24
4x6
J2
Pannello
C3
7407
IC3
Col.6 Col.5 Col.4 Col3 Col.2 Col.1
QTP 24
Titolo: QTP 24P
grifo®
Data: 16-07-1996
Rel. 1.1
Pagina : 1 of 2

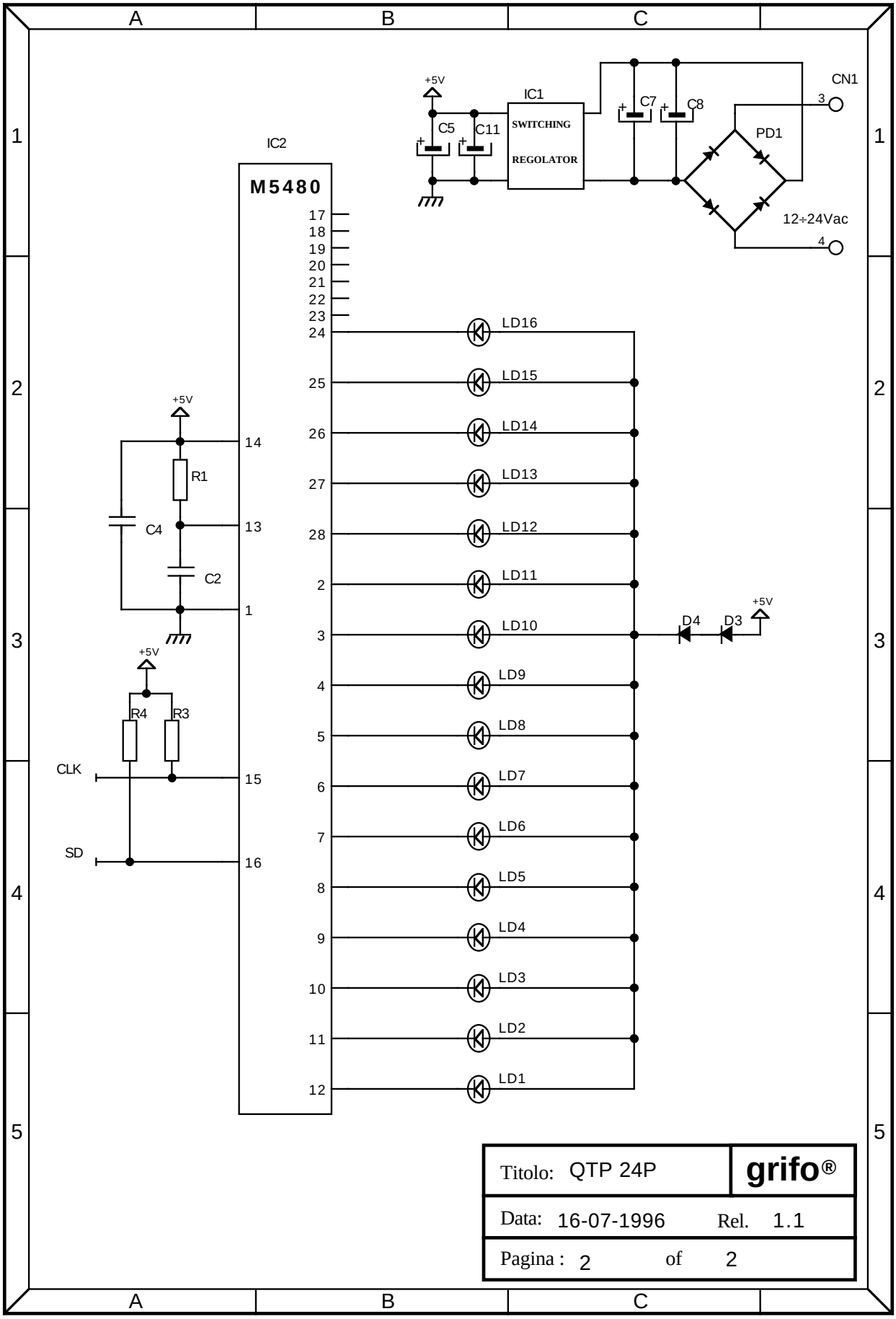


FIGURA 21: SCHEMA ELETTRICO QTP 24P

APPENDICE B: INDICE ANALITICO