

# GDOS

## 188

**grifo<sup>®</sup> Disk Operating System**  
**188 Family**

MANUALE UTENTE

**grifo<sup>®</sup>**

ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY

E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)

<http://www.grifo.it>

<http://www.grifo.com>

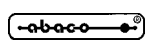
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



**GDOS 188**

Edizione 3.20

Rel. 22 Agosto 2001

, GPC<sup>®</sup>, grifo<sup>®</sup>, sono marchi registrati della ditta grifo<sup>®</sup>



# GDOS

## 188

### grifo<sup>®</sup> Disk Operating System 188 Family

#### MANUALE UTENTE

Il **GDOS 188** è un completo pacchetto software, realizzato dalla **grifo<sup>®</sup>**, per la schede basate sul microprocessore **80C188** e suoi derivati, della vasta famiglia **Intel 86**, appartenente al carteggio industriale **ABACO<sup>®</sup>**.

Questo manuale costituisce una completa guida utente per il corretto uso del **GDOS 188 (Grifo<sup>®</sup> Disk Operating System 188 family)**.

Il **GDOS 188** è composto da vari sottogruppi indipendenti e non, che insieme costituiscono un comodo ambiente di sviluppo. In particolare, oltre al **Sistema Operativo**, sono compresi il **Monitor/Debugger 188**, il compilatore **PASCAL 188**, in grado di generare codice romabile, ed il **TOOLS 188**, che facilita la gestione di alcune risorse di bordo della scheda.

**grifo<sup>®</sup>**

ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY

E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)

<http://www.grifo.it>

<http://www.grifo.com>

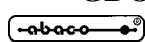
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



**GDOS 188**

Edizione 3.20

Rel. 22 Agosto 2001

, **GPC<sup>®</sup>**, **grifo<sup>®</sup>**, sono marchi registrati della ditta **grifo<sup>®</sup>**

## Vincoli sulla documentazione grifo® Tutti i Diritti Riservati

Nessuna parte del presente manuale può essere riprodotta, trasmessa, trascritta, memorizzata in un archivio o tradotta in altre lingue, con qualunque forma o mezzo, sia esso elettronico, meccanico, magnetico ottico, chimico, manuale, senza il permesso scritto della **grifo®**.

### IMPORTANTE

Tutte le informazioni contenute nel presente 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, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.

**grifo®** altresì 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 per **grifo®**.

Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

### LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:

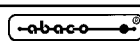


Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

### Marchi Registrati

 , GPC®, **grifo®** : sono marchi registrati della **grifo®**.

Altre marche o nomi di prodotti sono marchi registrati dei rispettivi proprietari.

# INDICE GENERALE

INTRODUZIONE .....	1
VERSIONI .....	1
GENERALITA' .....	2
MATERIALE NECESSARIO .....	3
SCHEDA DI CONTROLLO BASATA SUL 80C188 .....	3
EPROM O FLASH EPROM GDOS 188 .....	4
PERSONAL COMPUTER .....	4
CAVO DI COMUNICAZIONE SERIALE .....	4
SOFTWARE DI LAVORO .....	5
MANUALE UTENTE GDOS 188 .....	6
FIRMWARE DI PROGRAMMAZIONE .....	6
DISCO DISTRIBUITO .....	7
COME INIZIARE .....	8
UTILIZZO DEL PROGRAMMA GET188 .....	10
CARATTERISTICHE GENERALI .....	10
MATERIALE NECESSARIO .....	10
DESCRIZIONE DEL GET188 .....	11
INSTALLAZIONE .....	11
EDITOR .....	12
EMULAZIONE TERMINALE .....	14
ESECUZIONE DI UN COMPILATORE .....	20
CREAZIONE DI UN 'IMMAGINE BINARIA DELLA EPROM UTENTE .....	22
CARATTERISTICHE TECNICHE DEL GDOS 188 .....	24
VERSIONI GDOS 188 .....	25
DESCRIZIONE DEL GDOS 188 .....	26
WATCH DOG .....	26
LETTURA E SCRITTURA DALL'EEPROM SERIALE .....	26
GESTIONE DEL REAL TIME CLOCK .....	26
INTERFACCIE PER I/O DIGITALI .....	27
STAMPANTE LOCALE .....	27
JUMPER DI SETTAGGIO DEL GDOS 188 .....	28
SELEZIONE DELLA MODALITA' RUN / DEBUG .....	28
SELEZIONE DEL BAUD RATE DELLA LINEA DI CONSOLE .....	29
CONFIGURAZIONE DELLA SRAM SU GPC® 188D E GPC® 188F .....	29
ORGANIZZAZIONE DELLA MEMORIA .....	30
ORGANIZZAZIONE DELLA MEMORIA SU GPC® 188D .....	30
ORGANIZZAZIONE DELLA MEMORIA SU GPC® 188F .....	32

ORGANIZZAZIONE DELLA MEMORIA SU GPC® 883 .....	33
ORGANIZZAZIONE DELLA MEMORIA SU GPC® 884 .....	35
ORGANIZZAZIONE DELLA RAM DI LAVORO .....	37
ORGANIZZAZIONE DELLA EPROM O FLASH EPROM .....	39
ORGANIZZAZIONE DELLA EPROM .....	39
ORGANIZZAZIONE DELLA FLASH EPROM .....	41
GESTIONE DELLA FLASH EPROM .....	45
CONFIGURAZIONE SCHEDA .....	45
CARICAMENTO ED ESECUZIONE PROGRAMMA DI GESTIONE .....	45
FASI DEL PROGRAMMA DI GESTIONE .....	46
GESTIONE DELLA RAM-DISK .....	48
PROGRAMMA IN AUTORUN .....	49
MAPPAGGIO DELLE PERIFERICHE .....	50
MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188D .....	50
MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188F .....	52
MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 883 .....	54
MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 884 .....	56
<b>DESCRIZIONE DEL MONITOR / DEBUGGER 188 .....</b>	<b>57</b>
<b>FORMATO DEI PARAMETRI .....</b>	<b>57</b>
<b>VETTORI DI INTERRUPT .....</b>	<b>58</b>
<b>COMANDI DEL MONITOR / DEBUGGER 188 .....</b>	<b>58</b>
<b>ASSEMBLATORE IN LINEA .....</b>	<b>58</b>
<b>ESECUZIONE PROGRAMMA ALL'INDIRIZZO E0000H .....</b>	<b>59</b>
<b>COPIA DI UN BLOCCO DI MEMORIA .....</b>	<b>59</b>
<b>VISUALIZZAZIONE DI UN BLOCCO DI MEMORIA .....</b>	<b>60</b>
<b>INPUT DI UNA WORD .....</b>	<b>60</b>
<b>RIEMPIMENTO DI UN BLOCCO DI MEMORIA .....</b>	<b>60</b>
<b>ESECUZIONE DI UN BLOCCO DI PROGRAMMA .....</b>	<b>61</b>
<b>VISUALIZZAZIONE DELL'ELENCO DEI COMANDI .....</b>	<b>61</b>
<b>INPUT DI UN BYTE .....</b>	<b>61</b>
<b>ESECUZIONE PROGRAMMA ALL'INDIRIZZO F0000H .....</b>	<b>62</b>
<b>DEASSEMBLA UN BLOCCO DI MEMORIA .....</b>	<b>62</b>
<b>OUTPUT DI UN BYTE .....</b>	<b>62</b>
<b>ESECUZIONE DI UNA PROCEDURA .....</b>	<b>63</b>
<b>RIAVVIA IL GDOS 188 .....</b>	<b>63</b>
<b>VISUALIZZAZIONE O MODIFICA DEI REGISTRI .....</b>	<b>63</b>
<b>SOSTITUZIONE DI DATI IN MEMORIA .....</b>	<b>63</b>
<b>ESECUZIONE DI UNA O PIU' ISTRUZIONI .....</b>	<b>64</b>
<b>OUTPUT DI UNA WORD .....</b>	<b>65</b>
<b>VISUALIZZAZIONE IN FORMATO HEX INTEL .....</b>	<b>65</b>
<b>RICEZIONE DI UN FILE .....</b>	<b>65</b>
<b>TABELLA RIASSUNTIVA DEI COMANDI .....</b>	<b>66</b>
<b>DESCRIZIONE DEL PROGRAMMA TOOLS 188 .....</b>	<b>67</b>
<b>ESECUZIONE DEL PROGRAMMA TOOLS 188 .....</b>	<b>67</b>
<b>COMANDI DEL PROGRAMMA TOOLS 188 .....</b>	<b>68</b>
<b>SELEZIONE DRIVE E VISUALIZZAZIONE DIRECTORY .....</b>	<b>68</b>
<b>COPIA DI FILE .....</b>	<b>68</b>

CANCELLAZIONE DI FILE DA RAM-DISK .....	69
VARIAZIONE DEL NOME DI UN FILE SU RAM-DISK .....	69
FORMATTAZIONE DELLA RAM-DISK .....	69
SETTAGGIO DEL REAL TIME CLOCK .....	69
USCITA DAL PROGRAMMA TOOLS 188 .....	70
<b>FUNZIONI DEL SISTEMA OPERATIVO GDOS 188 .....</b>	<b>71</b>
<b>FUNZIONI DEL VETTORE DI INTERRUPT 3BH .....</b>	<b>71</b>
FUNZIONE 0: ACQUISIZIONE DI UN TASTO .....	72
FUNZIONE 1: STATO DELLA CONSOLE .....	72
FUNZIONE 2: INPUT DA CONSOLE .....	73
FUNZIONE 3: OUTPUT SU CONSOLE .....	73
FUNZIONE 4: STATO DELLA LINEA AUSILIARIA .....	73
FUNZIONE 15: OUTPUT SU CONSOLE DI "BELL" .....	73
FUNZIONE 16: OUTPUT SU CONSOLE DI "CR+LF" .....	73
FUNZIONE 20: POSIZIONAMENTO DEL CURSORE .....	74
FUNZIONE 21: CLEAR SCREEN .....	74
FUNZIONE 22: CLEAR END OF LINE .....	74
FUNZIONE 23: DELETE LINE .....	75
FUNZIONE 24: INSERT LINE .....	75
FUNZIONE 25: ALTA LUMINOSITA' .....	75
FUNZIONE 26: LUMINOSITA' NORMALE .....	76
FUNZIONE 31: ABILITA LA FORMATTAZIONE DELLA RAM-DISK .....	76
FUNZIONE 32: FORMATTAZIONE DELLA RAM-DISK .....	76
FUNZIONE 33: CONTROLLO DELLA RAM-DISK .....	77
FUNZIONE 34: LETTURA DI UN BYTE DALL'EEPROM .....	77
FUNZIONE 35: SCRITTURA DI UN BYTE NELL'EEPROM .....	78
FUNZIONE 36: SCRITTURA NELLA SRAM PROTETTA IN SCRITTURA .....	78
<b>TABELLA RIASSUNTIVA DELLE FUNZIONI DELL'INTERRUPT 3BH .....</b>	<b>79</b>
<b>FUNZIONI DEL VETTORE DI INTERRUPT 21H .....</b>	<b>80</b>
FUNZIONE 0: TERMINA L'ESECUZIONE DEL PROGRAMMA .....	80
FUNZIONE 3: INPUT DALLA LINEA AUSILIARIA .....	80
FUNZIONE 4: OUTPUT SU LINEA AUSILIARIA .....	81
FUNZIONE 5: OUTPUT SU STAMPANTE LOCALE .....	81
FUNZIONE 13: SVUOTA I BUFFER DEI FILE .....	81
FUNZIONE 14: SELEZIONE DEL DRIVE CORRENTE .....	82
FUNZIONE 25: ACQUISIZIONE DEL DRIVE CORRENTE .....	82
FUNZIONE 26: SETTA L'INDIRIZZO DELLA DTA .....	82
FUNZIONE 42: ACQUISIZIONE DELLA DATA .....	83
FUNZIONE 43: SETTAGGIO DELLA DATA .....	83
FUNZIONE 44: ACQUISIZIONE DELL'ORA .....	83
FUNZIONE 45: SETTAGGIO DELL'ORA .....	84
FUNZIONE 48: ACQUISIZIONE DEL NUMERO DI VERSIONE .....	84
FUNZIONE 54: ACQUISIZIONE DELLO SPAZIO LIBERO SU DISCO .....	84
FUNZIONE 60: MAKE DI UN FILE .....	85
FUNZIONE 61: OPEN DI UN FILE .....	86
FUNZIONE 62: CLOSE DI UN FILE .....	86
FUNZIONE 63: LEGGE DA UN FILE .....	87
FUNZIONE 64: SCRIVE SU UN FILE .....	87

<b>FUNZIONE 65: CANCELLA UN FILE .....</b>	<b>88</b>
<b>FUNZIONE 66: SPOSTA IL PUNTATORE SU UN FILE .....</b>	<b>88</b>
<b>FUNZIONE 68: CONTROLLO DELL'I/O PER I DISPOSITIVI .....</b>	<b>89</b>
<b>FUNZIONE 71: ACQUISIZIONE DELLA DIRECTORY .....</b>	<b>89</b>
<b>FUNZIONE 76: TERMINA L'ESECUZIONE DEL PROCESSO .....</b>	<b>90</b>
<b>FUNZIONE 78: TROVA LA PRIMA VOCE DI DIRECTORY .....</b>	<b>90</b>
<b>FUNZIONE 79: TROVA LA PROSSIMA VOCE DI DIRECTORY .....</b>	<b>91</b>
<b>FUNZIONE 86: CAMBIA IL NOME UN FILE .....</b>	<b>91</b>
<b>TABELLA RIASSUNTIVA DELLE FUNZIONI DELL'INTERRUPT 21H .....</b>	<b>92</b>
<b>APPENDICE A: SCHEMI INTERFACCIE I/O DIGITALI.....</b>	<b>A-1</b>
<b>APPENDICE B: INDICE ANALITICO .....</b>	<b>B-1</b>



# INDICE DELLE FIGURE

FIGURA 1: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE A SCATOLINO 16 VIE.....	4
FIGURA 2: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE PLUG 6 VIE .....	5
FIGURA 3: CONNETTORI SERIALI ED ACCESSORI.....	5
FIGURA 4: TABELLA 1 DELL SEQUENZE DI CONTROLLO DEL GET188 .....	17
FIGURA 5: TABELLA 2 DELLE SEQUENZE DI CONTROLLO DEL GET188 .....	18
FIGURA 6: CODICI DEI TASTI SPECIALI GESTITI DAL GET188 .....	19
FIGURA 7: CONNETTORI PER INTERFACCIE DI I/O .....	27
FIGURA 8: JUMPER DI SELEZIONE RUN/DEBUG .....	28
FIGURA 9: SELEZIONE BAUD RATE DELLA LINEA DI CONSOLE SU GPC® 188D E GPC® 188F .....	29
FIGURA 10: ORGANIZZAZIONE MEMORIA SU GPC® 188D .....	30
FIGURA 11: INDIRIZZI INIZIALI E FINALI DELLA SRAM IC9 SU GPC® 188D .....	31
FIGURA 12: ORGANIZZAZIONE MEMORIA SU GPC® 188F .....	32
FIGURA 13: ORGANIZZAZIONE MEMORIA SU GPC® 883 CON SRAM DA 128 KBYTES .....	33
FIGURA 14: ORGANIZZAZIONE MEMORIA SU GPC® 883 CON SRAM DA 512 KBYTES .....	34
FIGURA 15: ORGANIZZAZIONE MEMORIA SU GPC® 884 CON SRAM DA 128 KBYTES .....	35
FIGURA 16: ORGANIZZAZIONE MEMORIA SU GPC® 884 CON SRAM DA 512 KBYTES .....	36
FIGURA 17: ORGANIZZAZIONE DELLA RAM DI LAVORO .....	37
FIGURA 18: ORGANIZZAZIONE DELLA EPROM DI LAVORO .....	39
FIGURA 19: ORGANIZZAZIONE DELLA EPROM UTENTE .....	40
FIGURA 20: ORGANIZZAZIONE DELLA FLASH EPROM DI LAVORO .....	41
FIGURA 21: ORGANIZZAZIONE DELLA FLASH EPROM DI DEBUG .....	42
FIGURA 22: ORGANIZZAZIONE DELLA FLASH EPROM UTENTE .....	43
FIGURA 23: TABELLA 1 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188D .....	50
FIGURA 24: TABELLA 2 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188D .....	51
FIGURA 25: TABELLA 1 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188F .....	52
FIGURA 26: TABELLA 2 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188F .....	53
FIGURA 27: TABELLA 1 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 883 .....	54
FIGURA 28: TABELLA 2 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 883 .....	55
FIGURA 29: TABELLA DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 884 .....	56
FIGURA 30: TABELLA RIASSUNTIVA DEI COMANDI DEL MONITOR/DEBUGGER 188 .....	66
FIGURA 31: TABELLA DELLE FUNZIONI DELL'INTERRUPT 3BH .....	79
FIGURA 32: TABELLA 1 DELLE FUNZIONI DELL'INTERRUPT 21H .....	92
FIGURA 33: TABELLA 2 DELLE FUNZIONI DELL'INTERRUPT 21H .....	93
FIGURA 34: TABELLA 3 DELLE FUNZIONI DELL'INTERRUPT 21H .....	94
FIGURA A1: SCHEMA ELETTRICO DELLA SCHEDA IAC 01 .....	A-1



## INTRODUZIONE

L'uso di questo materiale é rivolto - IN VIA ESCLUSIVA - a personale specializzato.

Scopo di questo manuale é la trasmissione delle informazioni necessarie all'uso competente e sicuro dei prodotti. Esse sono il frutto di un'elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

I dati di seguito riportati sono destinati - IN VIA ESCLUSIVA - ad un utenza specializzata, in grado di interagire con i prodotti in condizioni di sicurezza per le persone, per la macchina e per l'ambiente, interpretando un'elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Per un corretto rapporto coi prodotti, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l'Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell'uso di tale ambiente di programmazione, é conveniente che l'utente - PRIMA DI COMINCIARE AD OPERARE - 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 all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

La **grifo®** non garantisce che questo software soddisfi le richieste dell'utente, che la produzione non cessi o sia priva di errori o che tutti gli eventuali errori siano corretti. La **grifo®** non é inoltre responsabile dei problemi causati dalle modifiche dell'hardware dei calcolatori e dei sistemi operativi che si possono verificare nel tempo.

Tutti i marchi registrati che compaiono nel presente manuale sono proprietà dei relativi costruttori.

## VERSIONI

Il presente manuale é riferito alle seguenti versioni:

- GDOS 188** per **GPC® 188D**: Versione **3.2** e successive
- GDOS 188** per **GPC® 188F**: Versione **3.3** e successive
- GDOS 188** per **GPC® 883**: Versione **3.3** e successive
- GDOS 188** per **GPC® 884**: Versione **3.2** e successive

La validità delle informazioni riportate é é quindi subordinata al numero di versione del firmware in uso e l'utente deve quindi sempre verificare la giusta corrispondenza tra le due indicazioni. Il numero di versione é scritto sull'etichetta presente nella EPROM o FLASH EPROM compresa nel pacchetto **GDOS 188**, oppure può essere ottenuta eseguendo lo stesso a bordo della scheda remota. Inoltre in questo manuale, sono presenti le informazioni relative al **GET188** versione **3.3** e successive; questo numero di versione é riportato nella finestra informativa, visualizzata all'avvio del programma e richiamabile con l'apposito comando.

## GENERALITA'

Il **GDOS 188** (**Grifo® Disk Operating System 188 family**) è un pacchetto software realizzato per tutte le schede basate sul microprocessore **80C188**, e suoi derivati, appartenente al carteggio industriale della **grifo®**.

Tale pacchetto consente di operare con questa scheda sfruttando un ambiente di lavoro molto evoluto, che non necessita nemmeno di una conoscenza approfondita di quello che è l'hardware che si utilizza.

Il **GDOS 188** è composto da vari sottogruppi indipendenti e non, che insieme costituiscono un comodo ambiente di sviluppo. In particolare, oltre al **Sistema Operativo**, sono compresi il **Monitor/Debugger 188**, il compilatore **PASCAL 188** in grado di generare codice romabile, ed il **TOOLS 188**, che facilita la gestione di alcune risorse di bordo della scheda remota, quali ad esempio la RAM-DISK, il Real-Time-Clock, ecc.

In questo manuale invece è riportata una descrizione completa di tutto il pacchetto, (escluso il compilatore **PASCAL 188**), seguendo una suddivisione logica relativa al significato dei singoli componenti.

Tramite il **GDOS 188** è possibile sfruttare il potente compilatore **PASCAL 188**, fornito nello stesso pacchetto software, ed avere così a disposizione molte risorse, normalmente non disponibili in altri ambienti di programmazione. E' infatti 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 188** è un semplice pacchetto software che può essere paragonato ad un elementare, ma efficace, sistema operativo per il settore industriale. Infatti come tutti i sistemi operativi, anche questo si occupa di effettuare un interfacciamento ad alto livello tra l'utente e l'hardware da utilizzare. Con tale pacchetto si ha quindi la possibilità di sfruttare tutte le risorse hardware del sistema, direttamente con istruzioni, comandi ad alto livello, senza doversi preoccupare di sviluppare firmware di gestione specifico. La notevole potenza e duttilità del **GDOS 188** permette inoltre di sfruttare al meglio le risorse di un calcolatore esterno, per semplificare notevolmente la fase di sviluppo e messa a punto dell'applicazione.

Vista la naturale evoluzione dei pacchetti software, si faccia sempre attenzione all'eventuale presenza del file "LEGGIMI.ITA" 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 è presente lo si deve esaminare, stampare ed allegare al presente manuale.

## MATERIALE NECESSARIO

Viene di seguito riportata una breve descrizione del materiale (hardware e software), necessario per operare con il **GDOS 188**:

### SCHEDA DI CONTROLLO BASATA SUL 80C188

Coincide con la scheda di controllo appartenente al carteggio industriale **grifo®**, basata sul microprocessore 80C188, o derivati, come: **GPC® 188D**, **GPC® 188F**, **GPC® 883**, **GPC® 884**, ecc.

#### **N.B.**

In questo manuale si utilizza la definizione **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 da almeno 128 KByte con pacchetto software **GDOS 188**  
almeno 32 KByte di SRAM.  
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 188** infatti, gestisce sempre con un'interfaccia ad alto livello, le seguenti sezioni della scheda:

- 1) *1 Linea seriale di console:* Indispensabile in tutti i casi in cui si utilizza il **GDOS 188** anche per lo sviluppo ed il debug dell'applicazione.
- 2) *1 Linea seriale ausiliaria:* Gestita ad alto livello la ricezione e la trasmissione di caratteri secondo un protocollo preimpostato.
- 3) *16 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**).
- 4) *SRAM non utilizzata:* Gestita ad alto livello come RAM DISK, in cui i dati sono salvati e prelevati sotto forma di files.
- 5) *Real Time Clock:* Gestito ad alto livello in termini di acquisizione e settaggio di data e ora.
- 6) *EEPROM:* Gestita ad alto livello in termini di lettura o scrittura di un singolo byte nelle varie locazioni di memoria.

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

## EPROM O FLASH EPROM GDOS 188

Il codice del sistema operativo viene sempre fornito in EPROM o FLASH EPROM 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.

## 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 successive. 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 moltissimo, 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 console della scheda remota, funziona da terminale con cui l'utente può interagire con il **GDOS 188** 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.

## CAVO DI COMUNICAZIONE SERIALE

Nel caso di utilizzo della linea seriale di console della scheda remota per lo sviluppo ed il debug del software di gestione, si deve provvedere a collegare il dispositivo di console in modo opportuno. Ricordando che la linea di consolle della scheda remota coincide sempre con la linea seriale A e che il **GDOS 188** necessita solo dei due segnali di ricezione e trasmissione (TxD e RxD), 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), come illustrato nelle figure seguenti.

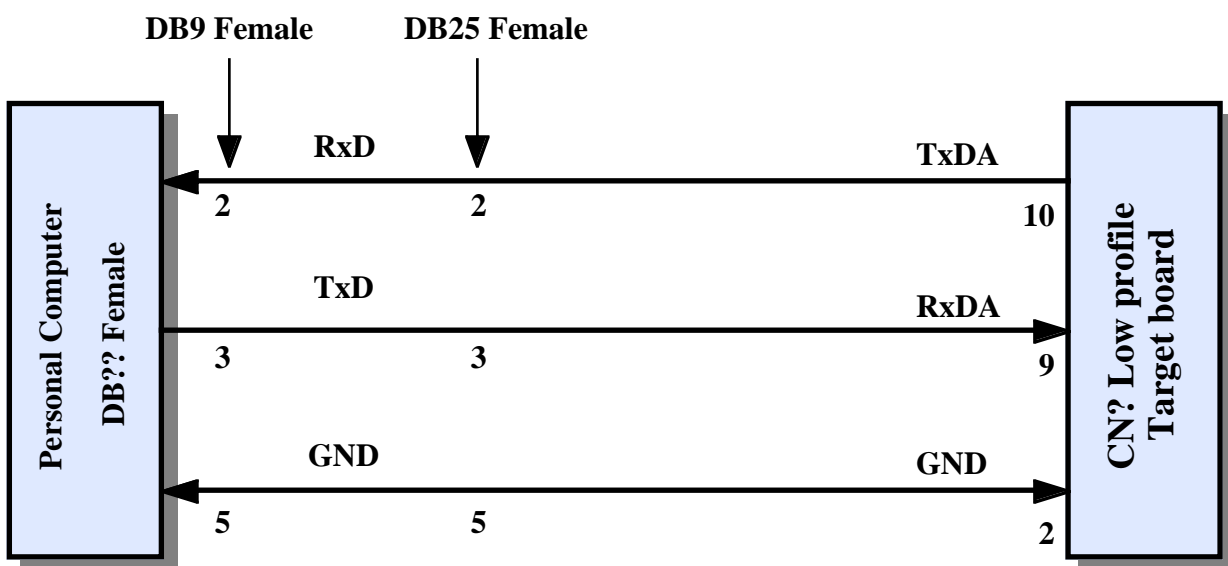


FIGURA 1: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE A SCATOLINO 16 VIE

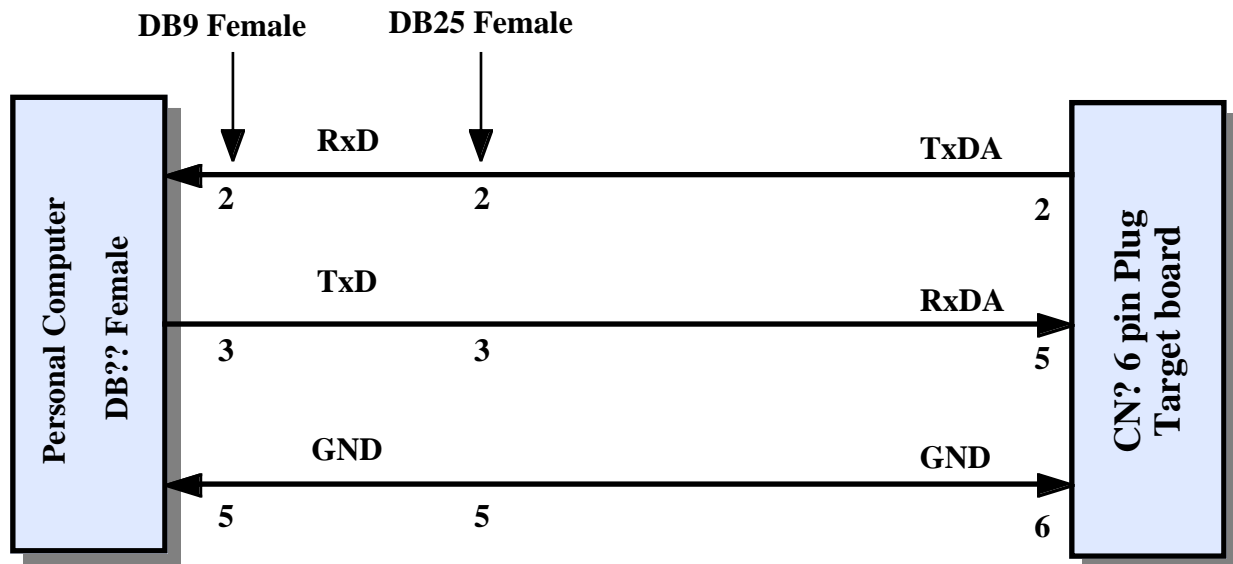


FIGURA 2: COLLEGAMENTO SERIALE TRA P.C. E SCHEDA CON CONNETTORE PLUG 6 VIE

Con le indicazioni **CN? scatolino scheda remota** e **CN? Plug 6 vie scheda remota** s'intendono i connettori standardizzati per la comunicazione seriale, della **grifo®**, descritti nel manuale tecnico della scheda remota. La tabella riportata nella figura seguente riporta i nomi di tali connettori ed i codici degli accessori (cavi, schede, ecc.) che la **grifo®** é in grado di offrire per facilitare e velocizzare la fase di connessione.

<i>SCHEDA REMOTA</i>	<i>CONNETTORE</i>	<i>CODICI ACCESSORI PER COLLEGAMENTO SERIALE</i>
<b>GPC® 188D</b>	CN1	FLT 16+16; NCS 01; CCR 25+25 o CCR 25+9
<b>GPC® 188F</b>	CN1	FLT 16+16; NCS 01; CCR 25+25 o CCR 25+9
<b>GPC® 883</b>	CN3A	CCR.PLUG25F o CCR.PLUG9F
<b>GPC® 884</b>	CN3A	CCR.PLUG25F o CCR.PLUG9F

FIGURA 3: CONNETTORI SERIALI ED ACCESSORI

**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 188**. Tra questi si ricordano i programmi di emulazione terminale intelligente (**GET188.EXE**), il programma di generazione della FLASH EPROM con il programma utente (**WFLASH.HEX**) ed i file, binario ed HEX Intel, del sistema operativo **GDOS 188**.

## MANUALE UTENTE GDOS 188

E' questo manuale e riporta tutte le informazioni tecniche relative al sistema operativo **GDOS 188**. In particolare si possono trovare connessioni hardware, sintassi dei comandi, descrizione delle funzioni, procedure e programmi di utility, uso della memoria, ecc.

### FIRMWARE 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 il firmware di programmazione del pacchetto **GDOS 188**, contenuto nella EPROM o FLASH EPROM montata sulla scheda remota ed è costituito da:

- *Sistema operativo GDOS 188*: Software che si occupa di effettuare un interfacciamento ad alto livello tra l'utente e l'hardware da utilizzare. Infatti tramite delle semplici chiamate alle varie funzioni del **GDOS 188**, sono accessibili sia le risorse interne della scheda remota, che quelle del P.C. utilizzato come terminale.
- *Monitor/Debugger 188*: Software che consente di effettuare delle operazioni che riguardano le risorse hardware di base della scheda remota e di debuggare, a livello Assembly, un programma utente. Alcune delle operazioni possibili sono input/output, verifica o sostituzione del contenuto della memoria, gestione registri della CPU, esecuzione passo-passo di un applicativo, settaggio break-point, ecc.
- *Compilatore PASCAL 188*: Ambiente di programmazione, che permette di realizzare e testare ogni tipo di programma che l'utente realizza, svilluppandolo direttamente a bordo scheda. Il programma utente può essere poi compilato per generare del codice roomabile da scaricare in EPROM.
- *TOOLS 188*: Software che facilita la gestione di alcune risorse di bordo della scheda remota; infatti è possibile formattare la RAM-Disk, copiare dei file da P.C. a RAM-Disk o viceversa, rinominare o cancellare dei file, settare il Real-Time-Clock, ecc.



## DISCO DISTRIBUITO

Il pacchetto software **GDOS 188**, oltre alla EPROM o FLASH EPROM, è normalmente composto da un disco che contiene tutta la struttura software necessaria per il lavoro.

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 188**, 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 188**, ovvero la struttura di lavoro principale:

- GET188.EXE -> Programma di editor ed emulazione terminale intelligente per **GDOS 188** da eseguire sul P.C.
- G188HELP.HLP -> File di help in linea per il **GET188.EXE**.

### Directory EPROM:

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 188** acquistato, EPROM o FLASH EPROM, saranno presenti alcuni dei file di seguito riportati.

- MON?????.BIN -> Immagine binaria del sistema operativo **GDOS 188** nella versione per la scheda **GPC®** ?????. In questo file é contenuto inoltre, il **Monitor/Debugger 188**. E' da utilizzare per creare la EPROM utente, mentre non é necessario se si utilizza una FLASH EPROM.
- 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.
- WFLASH.HEX -> Programma eseguibile sulla scheda remota, da utilizzarsi per la gestione della FLASH EPROM.
- TOOLS188.COM -> Immagine binaria del programma di utility **TOOLS 188**.
- PAS188\_L.BIN -> Immagine binaria del compilatore **PASCAL 188**, da utilizzare nel caso nel caso si debba inserire tale file in EPROM o FLASH EPROM a partire dall'indirizzo **E000:0000 Hex**.
- PAS188\_H.BIN -> Immagine binaria, parziale, del compilatore **PASCAL 188**, da utilizzare nel caso nel caso si debba inserire tale file in FLASH EPROM a partire dall'indirizzo **F000:0000 Hex**. Non é utilizzabile se si lavora con una EPROM.

### Directory ESEMPI:

Contiene una serie di programmi dimostrativi che illustrano come utilizzare le risorse di bordo della scheda; tale directory, a sua volta può contenere altre sottodirectory con programmi dimostrativi relativi ad applicazioni di uso generale.

## COME INIZIARE

In questo capitolo vengono descritte quelle che sono le operazioni da effettuare per un primo elementare utilizzo del pacchetto **GDOS 188**. 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.), assicurandosi, inoltre, che la EPROM o la FLASH EPROM sia montata sullo zoccolo opportuno, come indicato nel manuale tecnico della scheda stessa.
- 3) Assicurarsi, sulla scheda remota, che il jumper di Run/Debug sia non connesso e che la stessa sia configurata per comunicare con un Baud Rate di 115.2 KBaud.  
Normalmente la scheda viene venduta già configurata in queste modalità, comunque per effettuare tali verifiche fare riferimento ai paragrafi "SELEZIONE DELLA MODALITA' RUN / DEBUG" e "SELEZIONE DEL BAUD RATE DELLA LINEA DI CONSOLE".
- 4) Effettuare il collegamento seriale seguendo le indicazioni del paragrafo "CAVO DI COMUNICAZIONE SERIALE".
- 5) Accendere il Personal Computer.
- 6) 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 9.
- 7) 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 "DISCO DISTRIBUITO").
- 8) Portarsi nella directory di lavoro creata.
- 9) Installare il programma **GET188.EXE**, ovvero lanciare la routine di installazione, digitando dal prompt MS-DOS il comando **GET188 /I <Invio>**. 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), il nome dell'utente e della ditta che utilizzerà il programma **GET188**. Per maggiori informazioni si veda il paragrafo "INSTALLAZIONE".
- 10) Eseguire il programma di editor ed emulazione terminale "**GET188.EXE**", (vedere il paragrafo "DESCRIZIONE DEL GET188") ed attendere la comparsa della finestra di presentazione.
- 11) Chiudere la finestra di presentazione premendo <Invio> ed infine selezionare l'opzione "Terminal" (tasti ALT+T), 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:

F10 Menu | TERMINAL for Mon./Pas. 188 - GRIFO° Tel. +39-051-892052

- 12) Alimentare la scheda remota: in corrispondenza di questa operazione sul monitor del P.C. deve comparire un'indicazione relativa al compilatore PASCAL 188 eseguito dalla scheda remota, seguita dal prompt del sistema operativo, in particolare:

-----  
**PASCAL 188 Editor-Compiler** Version x.x grifo° Italian Technology  
-----

<Menù ed informazioni varie>

P188>

- 13) Lavorare con il compilatore PASCAL188 secondo le modalità illustrate nel manuale relativo. Per uscire dal **GET188** e ritornare al sistema operativo, premere i tasti "ALT+X". All'interno del compilatore PASCAL188, ad esempio, possono essere fornite le seguenti istruzioni:

P188>E <seguito dal nome del file come ad esempio PROVA.PAS>  
con cui si accede all'editor del programma PASCAL188 relativo nome inserito.

Program PROVA;

Var IND:Integer;

Begin

    For IND:= 1 To 10 Do  
        Writeln(IND);

End.

con cui si realizza un programma di conteggio da 1 a 10.

<Tasti CTRL+K> quindi <Tasto D>  
con cui si ritorna al menù principale

P188>R

con cui si compila in memoria, quindi si esegue il programma realizzato.

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

## UTILIZZO DEL PROGRAMMA GET188

Vengono di seguito riportate tutte le indicazioni che riguardano l'uso del programma di Editor e comunicazione intelligente per P.C. e del relativo programma di installazione.

### CARATTERISTICHE GENERALI

Il **GET188** (Grifo® Editor Terminal **188** family) si utilizza in congiunzione al pacchetto **GDOS 188**, 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: l'editor di programmi applicativi sviluppati dall'utente, e la simulazione di un terminale intelligente, che provvede a gestire tutte le funzioni di console come un semplice terminale, ma in più offre la possibilità di utilizzare le risorse di memoria di massa del P.C. per trasferire dei programmi alla scheda, oppure per salvare su disco un programma residente nella memoria locale della scheda. Inoltre lo stesso programma mette a disposizione due utility; la prima permette di eseguire un compilatore esterno, senza dover ritornare al sistema operativo. In questo modo è ad esempio possibile, editare un source Assembly con l'editor di bordo, compilarlo con il programma appropriato, senza ritornare al DOS, quindi inviare il relativo codice compilato al Monitor/Debugger 188 in esecuzione sulla scheda remota.

La seconda utility del **GET188** invece, permette di creare l'immagine della EPROM utente con il relativo programma in Auto-Run; il tutto in modo facile ed immediato, senza dover essere a conoscenza dei vari indirizzi fisici della EPROM, in cui andare a 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.

### MATERIALE NECESSARIO

Vengono di seguito riportate le caratteristiche fondamentali del programma **GET188.EXE** descritto, per quanto riguarda l'utilizzo delle risorse P.C.:

**Personal Computer:** IBM compatibile.

**Memoria RAM:** Minimo 640 KBytes.

**Sistema operativo:** MS-DOS versione 3.3 o successiva oppure Windows 95/98/ME.

**Monitor:** Colori o Bianco e nero

**Memorie di massa:** Drive ed eventualmente Hard Disk in qualsiasi formato gestito da MS-DOS.

**Seriale:** COM 1, 2, 3, 4 secondo specifiche V24.

**Mouse:** Microsoft compatibile con relativo driver installato.

## DESCRIZIONE DEL GET188

Il **GET188** 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.

Da notare che il P.C. che esegue il **GET188**, 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.

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

**GET188** <Invio>

da fornire direttamente al sistema operativo MS - DOS.

Una volta lanciato, il **GET188** 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 INVIO o facendo click con il mouse su "OK", tale finestra informativa scompare e diventa accessibile la finestra principale del programma; in questa sono disponibili sei menù a tendina provvisti di una serie di opzioni che vengono descritte nei seguenti paragrafi.

## INSTALLAZIONE

Prima di utilizzare il **GET188.EXE** si deve provvedere alla sua installazione. Per questo è stata realizzata un'apposita utility, che viene eseguita digitando, dal prompt del sistema operativo MS-DOS, il seguente comando:

**GET188** /I <Invio>

Questa utility d'installazione, una volta eseguita, 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 **GET188.EXE** non è già stato installato e se lo é vengono richiesti solo i primi tre; da questo si ricava che l'installazione in termini di nome dell'utente, può essere effettuata una sola volta.

Tali parametri sono quelli di default con cui parte il **GET188**, e possono essere variati in qualsiasi momento all'interno del programma stesso.

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.

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 **GET188.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 **GET188.EXE** comprende un potente e versatile Editor in grado di creare file ASCII direttamente utilizzabili dal **GDOS 188**. Esso 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.

Di seguito é riportato l'elenco dei comandi disponibili nella barra dei menù (attivabile con il tasto **F10**) quando il **GET188** é nella modalità di editor.

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Menu	F10	Attiva la barra dei menù per la selezione del comando desiderato

### Menù File

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
New	-	Apri un nuovo file da editare con il nome "Untitled"
Open ...	F3	Apri un file, ponendo il contenuto nella finestra di editor
Save	F2	Salva il contenuto della finestra di editor attiva in un file con il nome riportato sulla stessa finestra
Save as...	-	Salva il contenuto della finestra di editor attiva in un file con un nome specificato dall'utente
Change dir...	Alt+F5	Cambia la directory attuale del DOS
Dos shell	-	Esce temporaneamente in DOS, per eseguire altre applicazioni; il <b>GET188</b> rimane in memoria e non deve essere quindi rilanciato.
Exit to DOS	Alt+X	Esce definitivamente dal programma e ritorna al DOS.

### Menù Edit

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Undo	-	Elimina, se possibile, l'azione dell'ultima opzione eseguita
Cut	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
Clear	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

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
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	Ctrl+L	Ripete l'ultima opzione di "Find" o "Replace" eseguita

## Menù Windows

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Tile	-	Rappresenta tutte le finestre di editor aperte distribuendole sull'area di rappresentazione del monitor
Cascade	-	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 successiva 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

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Terminal	Alt+T	Attiva la modalità 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
Help	F1	Attiva la finestra di Help in linea del <b>GET188</b> .
Help Language..	-	Seleziona il linguaggio dell'Help in linea.
Information...	-	Visualizza la finestra di informazioni del programma <b>GET188</b> .

## Menù Utility

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Compiler...	-	Permette di eseguire un compilatore senza tornare al sistema operativo MS-DOS.
Compile again	Alt+F9	Permette di eseguire una nuova compilazione con i parametri precedentemente impostati.
Make EPROM..	-	Permette di creare l'immagine binaria del file da memorizzare in EPROM, con il programma utente in Auto-Run.

Nella descrizione delle opzioni precedentemente 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.).

Se si fa uso del mouse queste funzioni sono notevolmente semplificate, in quanto è sufficiente fare click sull'opzione del menù desiderata, senza premere alcun tasto da tastiera.

Una descrizione più approfondita delle modalità d'uso delle opzioni del **GET188** non viene qui riportata, in quanto è demandata direttamente all'Help in linea, disponibile all'interno del programma e richiamabile in qualsiasi momento con il tasto **F1**.

## 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 188**.

La comunicazione con la scheda remota viene gestita tramite una delle linee seriali del P.C. (COM) con il collegamento e protocollo indicato nei paragrafi "CAVO DI COMUNICAZIONE SERIALE" e "CARATTERISTICHE TECNICHE DEL **GDOS 188**".

Una volta attivata l'opzione "Terminal" del menù "Options" del **GET188**, viene settata la linea seriale attuale con il baud rate attuale e quindi rappresentata la finestra di emulazione terminale. Con linea seriale attuale e baud rate attuale s'intendono i valori settati dall'utente (tramite l'opzione "Serial port" del menù "Options") che inizialmente coincidono con i valori di default definiti in fase d'installazione del **GET188.EXE**. Al fine di velocizzare l'uso dell'emulazione terminale del **GET188** è 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 è completamente trasparente per l'utente e fa uso degli handshake software per regolamentare le comunicazioni veloci.

### N.B.

La gestione degli handshake software è associata al dato salvato nel **Flag di CONSOLE** (vedere il paragrafo "ORGANIZZAZIONE DELLA RAM DI LAVORO") con la seguente corrispondenza:

**Flag di CONSOLE (Ind. 00402H) = 55AA H** -> gestione handshake software disabilitata  
**Flag di CONSOLE (Ind. 00402H) ≠ 55AA H** -> gestione handshake software abilitata

Durante la comunicazione con il **GET188**, la gestione degli handshake software deve essere abilitata e per questa ragione di default il **GDOS 188** setta tale flag con un valore diverso da 55AA Hex. 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 settare diversamente questo flag.



Tutti i programmi realizzati con il **GDOS 188** 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 188**; quindi non deve essere rispettata nessuna sequenza di accensione o di spegnimento dei due sistemi.

### N.B.

Dal sistema operativo MS-DOS, è possibile eseguire il **GET188** ed entrare direttamente nella modalità di emulazione terminale intelligente; la sintassi da utilizzare per ottenere questo risultato è la seguente:

**GET188 /T <Invio>**

### Comandi dell'Emulazione Terminale:

In emulazione terminale sono disponibili una serie di comandi, disponibili nella barra dei menù (attivabile con il tasto **F10**), che aiutano l'utente nell'utilizzo del del firmware in esecuzione sulla scheda remota; di seguito ne è riportato l'elenco ed una breve descrizione:

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Menu	F10	Attiva la barra dei menù per la selezione del comando desiderato

### Menù File

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Change dir...	Alt+F5	Cambia la directory attuale del DOS
Dos shell	-	Esce momentaneamente dal programma per l'esecuzione di applicazioni DOS. Il <b>GET188</b> rimane in memoria e non deve essere quindi rilanciato.
Exit to DOS	Alt+X	Esce definitivamente dal programma e ritorna al DOS.

### Menù Options

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Editor	Alt+T	Attiva la modalità di editor del <b>GET188</b> .
Reset Terminal	Ctrl+Home	Cancella la finestra di emulazione terminale e resetta la comunicazione seriale con la scheda remota.
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
Help	F1	Attiva la finestra di Help in linea del <b>GET188</b> .
Help Language..	-	Seleziona il linguaggio dell'Help in linea.
Information...	-	Visualizza la finestra di informazioni del programma <b>GET188</b> .

## Menù Utility

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Compiler...	-	Permette di eseguire un compilatore senza tornare al sistema operativo MS-DOS.
Compile again	Alt+F9	Permette di eseguire una nuova compilazione con i parametri precedentemente impostati.
Make EPROM..	-	Permette di creare l'immagine binaria del file da memorizzare in EPROM, con il programma utente in Auto-Run.

Nel menù "Utility" sono inoltre presenti i due seguenti Sotto-menù:

### Sub-menù Transmit file to Monitor

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
HEX Intel file..	F8	Consente di trasferire al Monitor/Debugger, un file in formato HEX Intel, residente nella memoria di massa del P.C. Questa funzione utilizza il comando <b>X</b> del Monitor/Debugger, per maggiori chiarimenti si veda il capitolo "DESCRIZIONE DEL MONITOR/DEBUGGER".
Binary file..	Shift+F8	Consente di trasferire al Monitor/Debugger, un file in formato binario, residente nella memoria di massa del P.C. Il programma in questione, sarà memorizzato nella SRAM della scheda remota, a partire dall'indirizzo specificato nei registri <b>CS:IP</b> . Questa funzione utilizza il comando <b>X</b> del Monitor/Debugger, per maggiori chiarimenti si veda il capitolo "DESCRIZIONE DEL MONITOR/DEBUGGER".

### Sub-menù Send command to Pascal

<i>Opzione</i>	<i>Tasto</i>	<i>Funzione</i>
Save (^K D S)	F7	Invia al compilatore <b>PASCAL 188</b> , la sequenza di codici equivalente ai seguenti tasti: <b>CTRL+K D S</b> In questo modo é possibile, premendo il solo tasto <b>F7</b> , uscire dall'editor (equivalente ai tasti CTRL+K, D) e salvare il file corrente (equivalente al tasto S del menù principale).
Compile(^K D C) F9		Invia al compilatore <b>PASCAL 188</b> , la sequenza di codici equivalente ai seguenti tasti: <b>CTRL+K D C</b> In questo modo é possibile, premendo il solo tasto <b>F9</b> , uscire dall'editor (equivalente ai tasti CTRL+K, D) e compilare il file corrente (equivalente al tasto C del menù principale).
Run (^K D R)	Ctrl+F9	Invia al compilatore <b>PASCAL 188</b> , la sequenza di codici equivalente ai seguenti tasti: <b>CTRL+K D R</b> In questo modo é possibile, premendo i soli tasti <b>Ctrl+F9</b> , uscire dall'editor (equivalente ai tasti CTRL+K, D) ed eseguire il file corrente (equivalente al tasto R del menù principale).

Nella descrizione delle opzioni precedentemente 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 **GET188** non viene qui riportata, in quanto é demandata direttamente all'Help in linea, disponibile all'interno del programma e richiamabile in qualsiasi momento con il tasto **F1**.

### Sequenze di controllo dell'Emulazione Terminale:

L'emulazione terminale intelligente, del programma **GET188** riconosce alcune delle sequenze di controllo ricevute dalla linea seriale selezionata. Tali sequenze sono riportate nella seguente tabella:

<i>COMANDO</i>	<i>CODICE BYTE</i>	<i>CODICE ASCII</i>
<b>HOME</b>	01	SOH
<b>CURSOR LEFT</b>	08	BS
<b>CURSOR RIGHT</b>	06	ACK
<b>CURSOR DOWN</b>	10	LF
<b>CURSOR UP</b>	26	SUB
<b>CARRIAGE RETURN</b>	13	CR
<b>CARRIAGE RETURN + LINE FEED</b>	29	GS
<b>Posiz. assoluto cursore</b>	126 17 colonna riga	~ DC1 ASCII(colonna) ASCII(riga)
<b>CLEAR PAGE</b>	126 28	~ FS
<b>CLEAR LINE</b>	126 14	~ S0
<b>CLEAR END OF LINE</b>	126 15	~ SI
<b>CLEAR END OF PAGE</b>	126 16	~ DLE

**FIGURA 4: TABELLA 1 DELL SEQUENZE DI CONTROLLO DEL GET188**

<i>COMANDO</i>	<i>CODICE BYTE</i>	<i>CODICE ASCII</i>
<b>Cancellazione linea con SCROLL-UP</b>	126 19	~ DC3
<b>Inserimento nuova linea</b>	126 26	~ SUB
<b>Disattivazione cursore</b>	126 22	~ SYN
<b>Attivazione cursore a "Linea"</b>	126 20	~ SUB
<b>Attivazione cursore a "Blocco"</b>	126 21	~ NACK
<b>Impostazione attributo luminosità Normale</b>	126 25	~ EM
<b>Impostazione attributo Alta luminosità</b>	126 31	~ US
<b>Impostazione attributo di Reverse</b>	126 27	~ ESC
<b>BELL sonoro</b>	07	BEL

**FIGURA 5: TABELLA 2 DELLE SEQUENZE DI CONTROLLO DEL GET188**

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

126,17,20,10

Tutti i programmi realizzati possono utilizzare i comandi descritti, ottenendo così notevoli facilitazioni nella parte di programma che riguarda l'interfaccia utente.

**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 188**:

<i>TASTO</i>	<i>CODICI</i>	<i>CODICI HEX</i>
<b>Freccia UP</b>	0, 200	0, C8
<b>Freccia DOWN</b>	0, 208	0, D0
<b>Freccia LEFT</b>	0, 203	0, CB
<b>Freccia RIGHT</b>	0, 205	0, CD
<b>Page UP</b>	0, 201	0, C9
<b>Page DOWN</b>	0, 209	0, D1
<b>Home</b>	0, 199	0, C7
<b>End</b>	0, 207	0, CF
<b>Insert</b>	0, 210	0, D2
<b>Delete</b>	0, 211	0, D3

**FIGURA 6: CODICI DEI TASTI SPECIALI GESTITI DAL GET188**

Con questa caratteristica si ottengono notevoli facilitazioni soprattutto quando si utilizzano linguaggi di programmazione con un editor integrato (ad esempio il PASCAL 188), infatti ci si può spostare all'interno del programma applicativo in modo intuitivo, comodo e veloce.

## ESECUZIONE DI UN COMPILATORE

Mediante questa funzione é possibile eseguire la compilazione di un file qualsiasi, con il compilatore desiderato, senza dover tornare al sistema operativo.

In questo modo é possibile, ad esempio, generare un determinato source Assembly con l'editor, compilarlo con il programma opportuno, quindi trasferirlo al Monitor/Debugger 188 in esecuzione sulla scheda remota e provarlo. Il tutto senza mai abbandonare l'ambiente integrato del **GET188**, il quale permette di eseguire tutte le operazioni sopra citate.

Per eseguire un compilatore esterno attraverso l'apposita utility, l'utente deve svolgere le seguenti operazioni:

- 1) Assicurarsi che il compilatore, il file sorgente e l'eventuale linker o programma di conversione, siano presenti nei drive di cui dispone il P.C.
- 2) Eseguire l'apposita utility con il comando "Compiler.." del menù "Utility", in modo da aprire la finestra di compilazione.
- 3) Selezionare il programma compilatore da utilizzare.
- 4) Selezionare il file sorgente che deve essere compilato.
- 5) Introdurre, se necessari, i vari parametri richiesti dal compilatore.
- 6) Specificare se gli eventuali parametri devono essere posizionati prima o dopo il nome del file sorgente. Questo dipende dal tipo di compilatore usato, in quanto alcuni di essi richiedono i parametri quindi il nome del file sorgente, mentre altri il nome del file e quindi i parametri.
- 7) Introdurre, se necessaria, la riga di comando da eseguire al termine della compilazione. Questa funzione può rendersi utile se, ad esempio, dopo il compilatore é necessario eseguire un linker oppure un altro programma di conversione.
- 8) Confermare tutti i parametri inseriti ed eseguire la compilazione vera e propria.

Queste operazioni sono eseguibili attraverso i comandi della finestra di compilazione, i quali sono illustrati di seguito:

**Compiler** Permette di selezionare il nome del file compilatore che si desidera utilizzare. Il nome completo del relativo percorso, sarà visualizzato a fianco di tale comando; se non é stato ancora selezionato nessun file vi sarà l'indicazione "**Not Selected**".

**Source File** Permette di selezionare il nome del file sorgente da compilare. Il nome completo del relativo percorso, sarà visualizzato a fianco di tale comando; se non é stato ancora selezionato nessun file vi sarà l'indicazione "**Not Selected**".

**Parameters:** Permette di inserire una stringa di Max. 80 caratteri in cui specificare la serie di parametri da passare al compilatore. Se non si intende fornire nessun parametro bisogna lasciare tale riga vuota.

**Parameters position:** Indicazione della posizione dei parametri, prima del nome del file sorgente (*Before source file name*) o dopo (*After source file name*).

**Second command line:** Permette di inserire una stringa di Max. 70 caratteri, contenente una riga di comando MS-DOS, da eseguire al termine della compilazione.

Se questa stringa contiene dei caratteri, al termine della compilazione sarà fornita al sistema operativo, così come è stata inserita, senza nessuna variazione; in caso contrario si rientrerà nell'ambiente **GET188**.

Quando sono state settate tutte le varie opzioni si può eseguire la compilazione con "**Run**". In questa fase sarà visualizzata una schermata DOS con i messaggi dei vari programmi eseguiti. Tale operazione potrà poi essere rieseguita, sfruttando gli stessi parametri, con il comando "**Compile again**" (tasti **Alt+F9**) del menù "Utility".

### **N.B.**

Tutte le volte che viene eseguita una compilazione, il programma sorgente, se modificato con l'editor del **GET188**, deve essere salvato. Questo perché il compilatore preleva tale file da disco, quindi senza le eventuali ultime variazioni apportate.

### **Esempio**

La funzione di compilazione può anche essere usata per eseguire, ad esempio, la utility **GHEX2.COM**, contenuta nel disco distribuito.

Infatti se, ad esempio, si vuole convertire il file binario **PROVA.BIN**, in formato HEX Intel a partire dall'indirizzo **1000H**, sarà necessario eseguire le seguenti operazioni:

- 1) Selezionare con l'opzione "Compiler", il programma **GHEX2.COM**.
- 2) Selezionare con l'opzione "Source File", il file **PROVA.BIN**.
- 3) Inserire il parametro: **1000**.
- 4) Settare la posizione dei parametri dopo il nome del file sorgente.
- 5) Eseguire il programma confermando con "Run" i dati inseriti.
- 6) A questo punto il programma **PROVA.BIN** presente su disco, risulta convertito nel file HEX Intel **PROVA.HEX**.

## CREAZIONE DI UN 'IMMAGINE BINARIA DELLA EPROM UTENTE

Mediante questa funzione é possibile creare l'immagine binaria del file da memorizzare in EPROM, con il programma utente in autorun; il tutto in modo molto facile ed immediato, senza dover essere a conoscenza dei vari indirizzi fisici della EPROM in cui andare a scrivere i vari files.

Per la realizzazione dell'immagine della EPROM, attraverso l'apposita utility, l'utente deve svolgere le seguenti fasi:

- 1) Assicurarsi che il programma utente, l'eventuale file di dati e l'immagine binaria del **GDOS 188** (presente all'interno del disco distribuito), siano contenuti nei drive di cui dispone il P.C.
- 2) Eseguire l'utility con il comando "Make EPROM" del menù "Utility", in modo da aprire la relativa finestra.
- 3) Selezionare il file binario contenente l'immagine binaria del sistema operativo **GDOS 188 + Monitor/Debugger 188** che si desidera utilizzare. Tale file, denominato **MON?????.BIN** é contenuto nella directory EPROM del disco distribuito.
- 4) Selezionare il file eseguibile, relativo al programma utente. Questo può essere ottenuto, con il compilatore PASCAL 188; per far questo é necessario compilare il file sorgente, dopo aver selezionato l'opzione di compilazione su disco del menù "compiler Options"; in questo modo sarà creato un file con l'estensione **".COM"**, che può essere direttamente utilizzato.
- 5) Selezionare l'eventuale file contenente dati, messaggi od altro, utilizzato dal programma utente; questa operazione é opzionale. Il file in questione viene memorizzato nella EPROM all'indirizzo **11000H**, che il sistema operativo **GDOS 188** alloca a partire dalla locazione:

**F1000H = F100:0000**

Quindi i dati contenuti in tale file, saranno utilizzabili dal programma utente, mediante delle operazioni di lettura in memoria, a partire da tale indirizzo.

Per ulteriori chiarimenti si veda il paragrafo "ORGANIZZAZIONE DELLA EPROM O FLASH EPROM".

- 6) Introdurre il nome del file binario, generato dall'utility, con l'immagine della EPROM da programmare; l'eventuale area non utilizzata é riempita con dati non significativi (OFF Hex).
- 7) Conferare tutti i parametri inseriti ed eseguire la creazione vera e propria del file con l'immagine binaria della EPROM.
- 8) Uscire dal **GET188** riportandosi in ambiente DOS, in cui é possibile verificare l'effettiva presenza del file creato.
- 9) Scrivere il file binario ottenuto, all'interno di una EPROM 27c010 (1 Mbit x 8). Tale file deve essere programmato a partire dall'indirizzo **0000 Hex**, quindi se il programmatore richiede il formato **HEX Intel**, provvedere a trasformare il file binario in questione nell'equivalente HEX, specificando questo indirizzo come offset. Per ottenere questo scopo può essere utilizzato il programma **GHEX2.COM**, fornito all'interno del disco distribuito.



La EPROM così ottenuta può essere installata sulla scheda remota; questa all'atto del reset o del Power-ON esegue in modalità Auto-Run, il programma precedentemente selezionato.

Le operazioni 3-7 precedentemente descritte sono eseguibili attraverso i comandi della finestra in questione, i quali sono illustrati di seguito:

**Monitor File** Permette di selezionare il nome dell'immagine binaria del sistema operativo. Il nome completo del relativo percorso, sarà visualizzato a fianco del comando; se non è stato ancora selezionato nessun file vi sarà l'indicazione "**Not Selected**".

**Program File** Permette di selezionare il nome del file eseguibile, relativo al programma utente. Il nome completo del relativo percorso, sarà visualizzato a fianco del comando; se non è stato ancora selezionato nessun file vi sarà l'indicazione "**Not Selected**".

**Data File** Permette di selezionare il nome di un eventuale file di dati, messaggi od altro, utilizzato dal programma utente. Il nome completo del relativo percorso, sarà visualizzato a fianco del comando; se non è selezionato nessun file vi sarà l'indicazione "**Not Selected**".

**BIN EPROM File** Permette di selezionare il nome del file Binario creato, da memorizzare in EPROM. Il nome completo del relativo percorso, sarà visualizzato a fianco del comando; se non è stato ancora selezionato nessun file vi sarà l'indicazione "**Not Selected**".

Una volta che sono state settate tutte le varie opzioni si può eseguire il comando "**Make**". In questa fase sarà visualizzata una finestra con l'indicazione percentuale di lavoro svolto ed una linea di stato in cui vengono riportati i messaggi relativi alle varie operazioni effettuate.

#### **N.B.**

La selezione del file di dati è opzionale; se questa è già stata effettuata e la si vuole annullare, è necessario entrare nel menu di selezione con il comando "Data File", quindi selezionare l'opzione "Clear".

## CARATTERISTICHE TECNICHE DEL GDOS 188

Vengono di seguito riportate tutte le caratteristiche tecniche del pacchetto **GDOS 188**; 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 **L**  
 Drive **M** di RAM-Disk tramite la SRAM di bordo

Tutti questi drive possono avere formati variabili ed in caso della RAM-Disk viene gestita anche la circuiteria di protezione in scrittura.

- 2) Comunicazione:** Linea di console -> Linea seriale A  
 Linea ausiliaria -> Linea seriale B

Tali linee sono inizializzate e gestite dal **GDOS 188** con il seguente protocollo di comunicazione:

*Linea di console:* Baud Rate = Schede **GPC® 188F** e **GPC® 188D**: selezionabile tra  
 19200, 38400, 57600 e 115200 Baud  
 Schede **GPC® 883** e **GPC® 884**: 115200 Baud  
 Parità = Nessuna  
 Stop Bit = 1  
 Bit x Chr. = 8  
 Handshake = Nessuno

*Linea ausiliaria:* Baud Rate = 9600 Baud  
 Parità = Nessuna  
 Stop Bit = 1  
 Bit x Chr. = 8  
 Handshake = Nessuno

- 3) SRAM:** Scheda **GPC® 188D**: minimo 32 KBytes  
 Schede **GPC® 188F**, **GPC® 883** e **GPC® 884**: minimo 128 KBytes
- 4) Stampante:** Parallela collegata al connettore di I/O della scheda remota
- 5) Watch Dog:** Retriggerato in fase di formattazione e lettura da RAM-Disk e durante l'attesa del Time-Out della stampante.
- 5) Real Time Clock:** Gestito ad alto livello tramite funzioni del **GDOS 188**
- 6) EEPROM:** Gestita ad alto livello tramite funzioni del **GDOS 188**

## VERSIONI GDOS 188

Di base sono disponibili le seguenti versioni del pacchetto software:

### **GDOS xxxx**

Ambiente di sviluppo **GDOS 188** per la scheda **GPC® xxxx** composto da: EPROM da 128 KBytes, manuale utente e disco.

### **FGDOS xxxx**

Ambiente di sviluppo **GDOS 188** per la scheda **GPC® xxxx** composto da: FLASH EPROM da 128 KBytes, manuale utente e disco.

### **FGD xxxx**

FLASH EPROM da 128 KByte per la scheda **GPC® xxxx**, con il **GDOS 188**.

Le sigle sopra riportate si adattano alla scheda con frequenza di clock di base (20 MHz per **GPC® 188F/D**, 26 MHz per **GPC® 883** e **GPC® 884**); qualora si intenda utilizzare il **GDOS 188** con frequenze di clock superiori, deve utilizzare le apposite versioni identificate dal suffisso **.xxM** (ad esempio **.40M** per le **GPC® 883** e **GPC® 884** a 40 MHz di clock). Si ricorda inoltre, che le sigle sopra riportate possono essere utilizzate direttamente per gli eventuali ordini.

Come ogni software e firmware, anche il **GDOS 188** 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 seguito viene quindi riportata una breve descrizione delle modifiche che il pacchetto di sviluppo ha subito a seconda del numero di versione:

- Ver. 2.2 -> Prima versione.
- Ver. 3.1 -> Aggiunta gestione FLASH EPROM; modificato protocollo seriale per la comunicazione con il **GET188**.
- Ver. 3.2 -> Aggiunta gestione EEPROM e Watch Dog; variate le priorità di Autorun; aggiunta gestione **GPC® 884**.
- Ver. 3.3 -> Aggiunta gestione **GPC® 883**.

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

## DESCRIZIONE DEL GDOS 188

Il **GDOS 188** é a tutti gli effetti un sistema operativo che di effettuare un interfacciamento ad alto livello tra l'utente e l'hardware da utilizzare. Infatti tramite delle semplici chiamate alle varie funzioni, sono accessibili sia le risorse interne della scheda remota, che quelle del P.C. utilizzato come terminale. Viene di seguito riportata una descrizione del modo di utilizzo del pacchetto **GDOS 188** sulla scheda remota. In particolare vengono riportate tutte le indicazioni che riguardano la gestione ad alto livello delle varie periferiche, l'organizzazione della memoria, il mappaggio dei vari registri e la gestione dei jumper di configurazione e dei programmi in Autorun.

### WATCH DOG

Per tutte le schede remote provviste di circuiteria di watch dog, il **GDOS 188** provvede ad effettuare un'operazione di retrigger durante tutte le operazioni che richiedono un tempo di esecuzione, che può essere superiore al tempo di intervento del watch dog stesso.

Questo avviene durante le seguenti operazioni: fase di boot del **GDOS 188**, formattazione della RAM-Disk, lettura di un file da RAM-Disk e attesa del time-out nella gestione della stampante.

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 fra quelle precedentemente elencate.

### LETTURA E SCRITTURA DALL'EEPROM SERIALE

Il **GDOS 188** mette a disposizione due funzioni, associate all'**INT 3B Hex**, che permettono di leggere e/o scrivere un byte, in un determinato indirizzo della EEPROM seriale della scheda remota.

Tali funzioni, la cui descrizione é riportata nel capitolo "FUNZIONI DEL SISTEMA OPERATIVO **GDOS 188**", sono richiamabili tramite comandi ad alto livello del compilatore PASCAL 188 e permettono di semplificare la gestione del dispositivo da parte dell'utente e consentono di ridurre la quantità di codice generato.

Il **GDOS 188** gestisce sei size di EEPROM serili che sono: 2 KBit (256 Bytes), 4 KBit (512 Bytes), 8 KBit (1024 Bytes), 16 KBit (2048 Bytes), 32 KBit (4096 Bytes) e 64 KBit (8192 Bytes); da notare che non viene effettuata nessuna verifica sull'indirizzo da leggere o scrivere in relazione alla quantità di EEPROM installata, quindi l'utente deve preoccuparsi di non fornire dei valori al di fuori dell'ultima locazione disponibile.

### GESTIONE DEL REAL TIME CLOCK

Il **GDOS 188** mette a disposizione alcune funzioni, associate all'**INT 21 Hex**, che permettono di acquisire e settare la data e l'ora del Real Time Clock della scheda remota.

Tali funzioni, la cui descrizione é riportata nel capitolo "FUNZIONI DEL SISTEMA OPERATIVO **GDOS 188**", sono richiamabili tramite comandi ad alto livello del compilatore PASCAL 188 e permettono di semplificare la gestione del dispositivo da parte dell'utente e consentono di ridurre la quantità di codice generato.

## 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 interfacce con lo stesso connettore standardizzato di I/O, che possono essere collegate direttamente alla scheda remota di controllo con un semplice flat. Tra queste interfacce si possono ricordare schede didattiche, interfacce di I/O di potenza per il collegamento con i segnali del campo, schede con memoria di massa, interfacce operatore, ecc. Per alcune di queste schede il **GDOS 188** si occupa della loro gestione ad alto livello, in modo che l'utente le possa usare senza preoccuparsi dell'hardware di queste interfacce, 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 interfacce successivamente descritte e del relativo cavo di connessione.

<i>SCHEDA REMOTA</i>	<i>CONNETTORE</i>	<i>CAVO DI CONNESSIONE</i>
<b>GPC® 188D</b>	CN2	<b>FLT 20+20</b>
<b>GPC® 188F</b>	CN2	<b>FLT 20+20</b>
<b>GPC® 883</b>	CN7	<b>FLT 20+20</b>
<b>GPC® 884</b>	CN5	<b>FLT 26+20</b>

**FIGURA 7: CONNETTORI PER INTERFACCIE DI I/O**

Naturalmente una sola delle interfacce successivamente descritte può essere collegata e quindi usata con il **GDOS 188** ed in caso di utilizzo di una delle interfacce le sedici linee di I/O del connettore riportato in tabella non sono più a disposizione utente. Il **GDOS 188** 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 varie schede di interfaccia del carteggio **grifo®**, 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 7. 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 si può utilizzare il comando `Write(LST,...)` del compilatore PASCAL 188, oppure si possono chiamare direttamente le funzioni del **GDOS 188**, come illustrato in seguito.

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 188** non interviene sulle linee di I/O. La procedura di stampa del **GDOS 188** non è sospensiva: se la stampante non è pronta, dopo un timeout di **2 secondi**, la stessa procedura viene abbandonata; non è inoltre possibile avere informazioni sullo stato della stampante.

## JUMPER DI SETTAGGIO DEL GDOS 188

Il sistema operativo **GDOS 188** all'atto del reset o power-on, verifica lo stato di una serie di jumper presenti sulla scheda remota, per settare opportunamente le varie modalità di funzionamento.

In questo modo l'utente può comodamente selezionare la modalità da lui prescelta, in qualsiasi condizione operativa, senza modificare il software in uso.

Infatti mediante questi jumper é possibile selezionare la modalità Run/Debugger, settare il Baud Rate della linea di console e configurare RAM-Disk. Di seguito é riportato il significato di tali jumper, le loro possibili connessioni e le configurazioni relative. Per l'individuazione dei jumper a bordo scheda remota si faccia riferimento alle varie figure riportate nel relativo MANUALE TECNICO.

### SELEZIONE DELLA MODALITA' RUN / DEBUG

Mediante un jumper presente a bordo della scheda remota, é possibile selezionare la modalità di funzionamento del sistema operativo **GDOS 188**, fra i modi **Run** o **Debug**; questo avviene all'atto del reset o del power-on, quando viene verificato lo stato di tale jumper; in particolare:

*Modalità RUN:* Il programma utente, oppure il compilatore PASCAL 188 (nel caso della struttura di lavoro), viene eseguito in modo automatico.

*Modalità DEBUG:* Anche se presente un programma utente, oppure il compilatore PASCAL 188, viene eseguito il programma di **Monitor/Debugger 188**.

La selezione del modo di funzionamento, é di particolare interesse quando si deve intervenire su un sistema già installato. Abilitando infatti la modalità Debug, 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.

Nella figura seguente viene riportato il nome del jumper in questione, in relazione alla scheda utilizzata.

<i>SCHEDA REMOTA</i>	<i>JUMPER RUN/DEBUG</i>
<b>GPC® 188D</b>	J18
<b>GPC® 188F</b>	J18
<b>GPC® 883</b>	J1
<b>GPC® 884</b>	J1

**FIGURA 8: JUMPER DI SELEZIONE RUN/DEBUG**

Le modalità di connessione del jumper, in relazione al modo di funzionamento, sono le seguenti:

*Jumper J??*      *Connesso*      ->    *Modalità DEBUG*  
                          *Non connesso*      ->    *Modalità RUN*

Per ulteriori informazioni fare riferimento al paragrafo "PROGRAMMA IN AUTORUN", ed al capitolo "DESCRIZIONE DEL MONITOR/DEBUGGER 188".

## SELEZIONE DEL BAUD RATE DELLA LINEA DI CONSOLE

Mediante due jumpers, presenti solo sulle schede **GPC® 188D** e **GPC® 188F** e denominati **J16** e **J17**, é possibile selezionare il Baud Rate della linea di console del **GDOS 188** (linea seriale A della scheda remota); questo avviene all'atto del reset o del power-on, quando viene verificato lo stato di tali jumpers.

In questo modo l'utente può comodamente settare uno dei vari Baud Rate disponibili per la linea di console, in funzione del terminale o del P.C. utilizzato, senza modificare il software in uso.

Le possibili configurazioni ed i relativi Baud Rate selezionabili, sono riportati nella seguente tabella.

Jumper J16	Jumper J17	BAUD RATE
Non connesso	Non connesso	<b>115200 Baud</b>
Connesso	Non connesso	<b>57600 Baud</b>
Non connesso	Connesso	<b>38400 Baud</b>
Connesso	Connesso	<b>19200 Baud</b>

**FIGURA 9: SELEZIONE BAUD RATE DELLA LINEA DI CONSOLE SU GPC® 188D E GPC® 188F**

### N.B.

Questa particolare modalità di funzionamento non é disponibile quando si utilizza il **GDOS 188** con le **GPC® 883** e **GPC® 884**, in quanto la linea di console di tali schede viene sempre inizializzata con il Baud Rate di 115200 Baud.

Per ulteriori chiarimenti, sul protocollo di comunicazione della linea di console, si faccia riferimento al capitolo "CARATTERISTICHE TECNICHE DEL **GDOS 188**".

## CONFIGURAZIONE DELLA SRAM SU GPC® 188D E GPC® 188F

Mediante il jumper **J2**, presente solo sulle **GPC® 188D** e **GPC® 188F**, é possibile abilitare o disabilitare la sezione di protezione della scrittura, nella SRAM montata sullo zoccolo IC9 della scheda in questione. Il dispositivo é configurato come RAM-Disk, nel caso in cui sia attivata tale circuiteria, in caso contrario risulta essere della normale SRAM, gestita dal **GDOS 188** come RAM di lavoro.

In particolare, sono possibili le seguenti configurazioni:

<b>Jumper J2</b>	<i>Connesso</i>	->	<i>IC9 = RAM-Disk</i>
	<i>Non connesso</i>	->	<i>IC9 = RAM di lavoro</i>

Per ulteriori chiarimenti si faccia riferimento ai prossimi paragrafi "ORGANIZZAZIONE DELLA MEMORIA" e "GESTIONE DELLA RAM-DISK".

**ORGANIZZAZIONE DELLA MEMORIA**

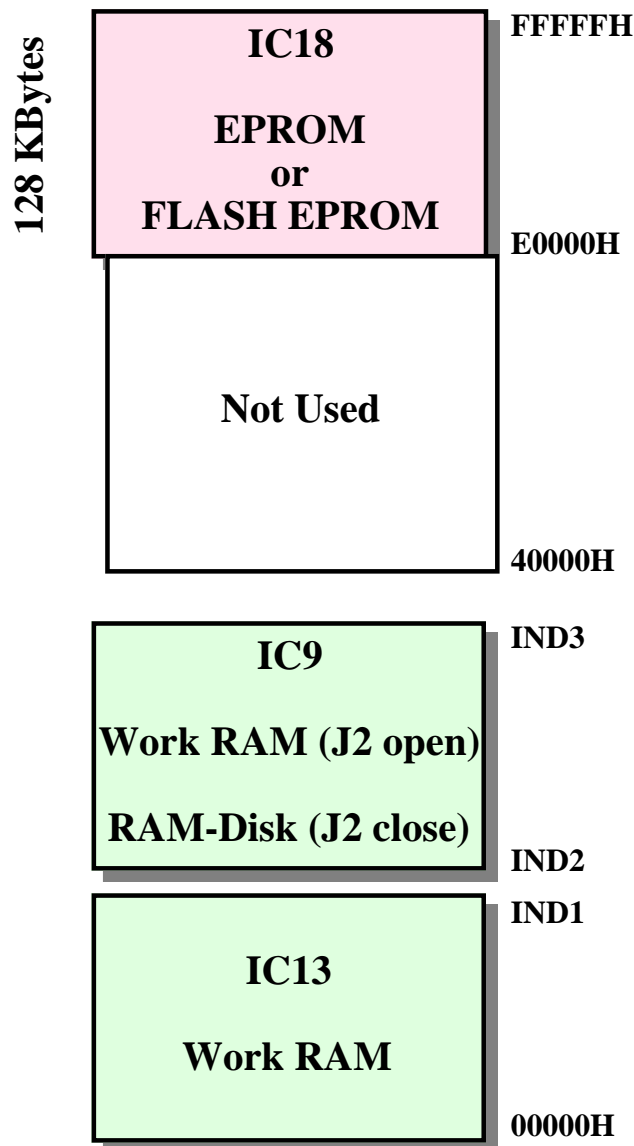
Il **GDOS 188** si occupa di organizzare tutta la memoria presente sulla scheda remota, rendendola disponibile all'utente tramite una serie di comandi ad alto livello. Per questo non é necessario preoccuparsi della configurazione hardware della scheda, ma é sufficiente rispettare le indicazioni fornite in questo paragrafo, utilizzando quelle che sono le aree consentite.

Nel caso che si utilizzi il compilatore PASCAL 188, questa verifica viene eseguita direttamente dal compilatore stesso.

Nei paragrafi seguenti viene riportata l'organizzazione della memoria relativamente alle varie schede che utilizzano il **GDOS 188**.

**ORGANIZZAZIONE DELLA MEMORIA SU GPC® 188D**

All'atto del reset o power-on, il **GDOS 188** verifica la quantità di SRAM e l'abilitazione o meno della RAM-Disk (jumper **J2**); quindi viene organizzata la memoria disponibile come illustrato di seguito.



**FIGURA 10: ORGANIZZAZIONE MEMORIA SU GPC® 188D**



Nella mappa di memoria riportata in figura, sono stati riportati alcuni indirizzi in modo generico; questi infatti, variano in funzione del tipo di dispositivo montato sul relativo zoccolo e dalla abilitazione, o meno, della RAM-Disk (jumper **J2** come descritto precedentemente).

Di seguito viene riportato il significato di tali indicazioni generiche.

**IND1:** *Indirizzo finale della RAM di lavoro su IC13*

Indica l'ultimo indirizzo disponibile, relativo alla quantità di SRAM montata sullo zoccolo IC13 della **GPC® 188D**; esso può avere i seguenti valori:

**SRAM IC13 da 32KByte** -> IND1=07FFFH (0700:0FFF)

**SRAM IC13 da 128 KByte** -> IND1=1FFFFH (1F00:0FFF)

**IND2 e IND3:** *Indirizzi iniziale e finale della RAM di lavoro o RAM-Disk su IC9*

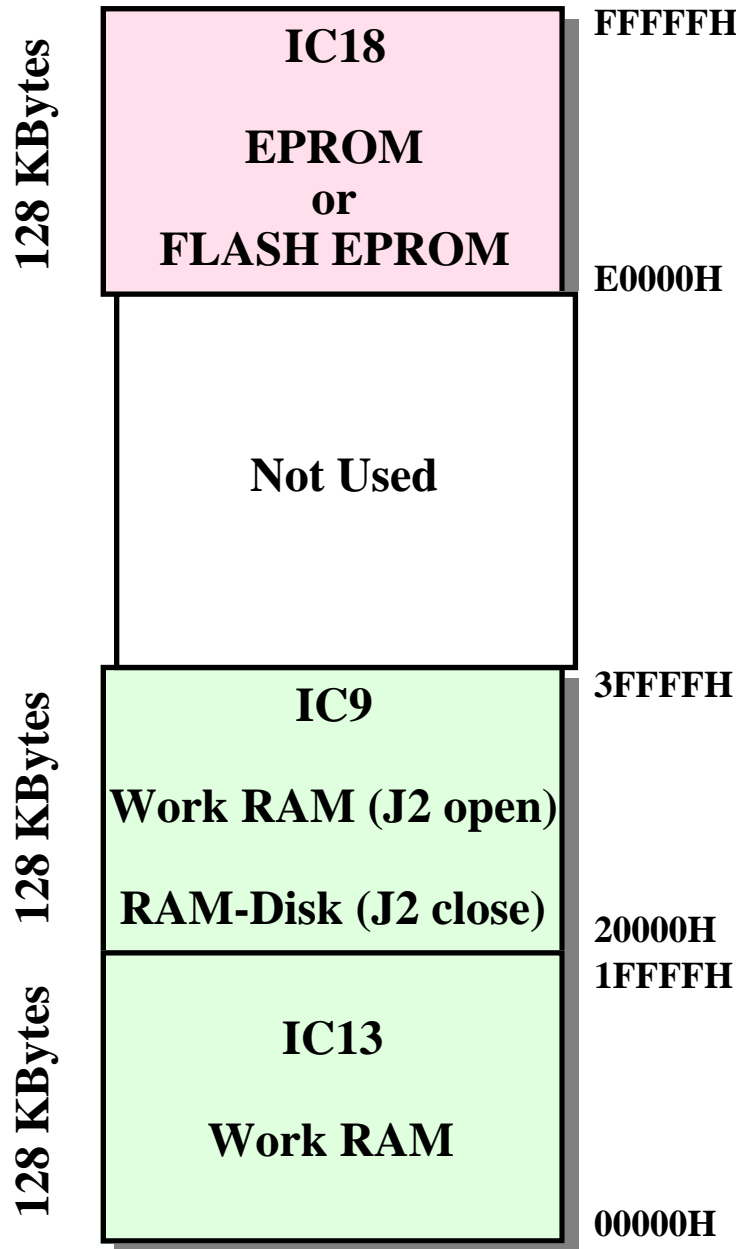
Questi indirizzi, variano in funzione della quantità di SRAM, montata sugli zoccoli IC9 e IC13 della scheda e dalla abilitazione, o meno, della RAM-Disk. Per determinare i valori relativi si faccia riferimento alla seguente tabella.

SRAM su IC13	SRAM su IC9	Jumper J2	IND2	IND3
32 KByte	32 KByte	Non connesso	08000H 0800:0000	0FFFFH 0F00:0FFF
32 Kbyte	32 KByte	Connesso	20000H 2000:0000	27FFFH 2700:0FFF
32 Kbyte	128 Kbyte	Non connesso	08000H 0800:0000	27FFFH 2700:0FFF
32 Kbyte	128 Kbyte	Connesso	20000H 2000:0000	3FFFFH 3F00:0FFF
128 Kbyte	32 KByte	Non connesso	20000H 2000:0000	27FFFH 2700:0FFF
128 Kbyte	32 KByte	Connesso	20000H 2000:0000	27FFFH 2700:0FFF
128 Kbyte	128 Kbyte	Non connesso	20000H 2000:0000	3FFFFH 3F00:0FFF
128 Kbyte	128 Kbyte	Connesso	20000H 2000:0000	3FFFFH 3F00:0FFF

**FIGURA 11: INDIRIZZI INIZIALI E FINALI DELLA SRAM IC9 SU GPC® 188D**

## ORGANIZZAZIONE DELLA MEMORIA SU GPC® 188F

All'atto del reset o power-on, il **GDOS 188** verifica la quantità di SRAM e l'abilitazione o meno della RAM-Disk (jumper **J2**); quindi viene organizzata la memoria disponibile come illustrato di seguito.

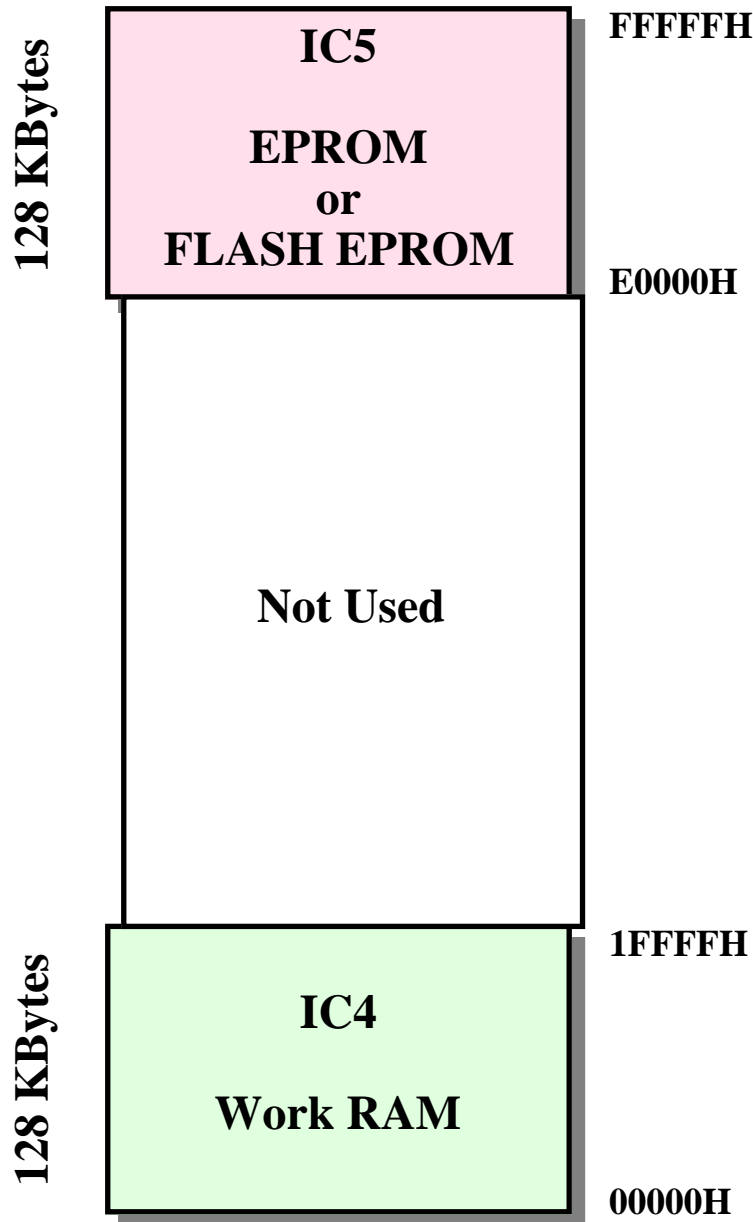


**FIGURA 12: ORGANIZZAZIONE MEMORIA SU GPC® 188F**

Come si può notare dalla figura, sono disponibili 256 KBytes di RAM di lavoro se il jumper **J2** non é connesso; viceversa vi sono 128 KBytes di RAM di lavoro e 128 KBytes di RAM-Disk.

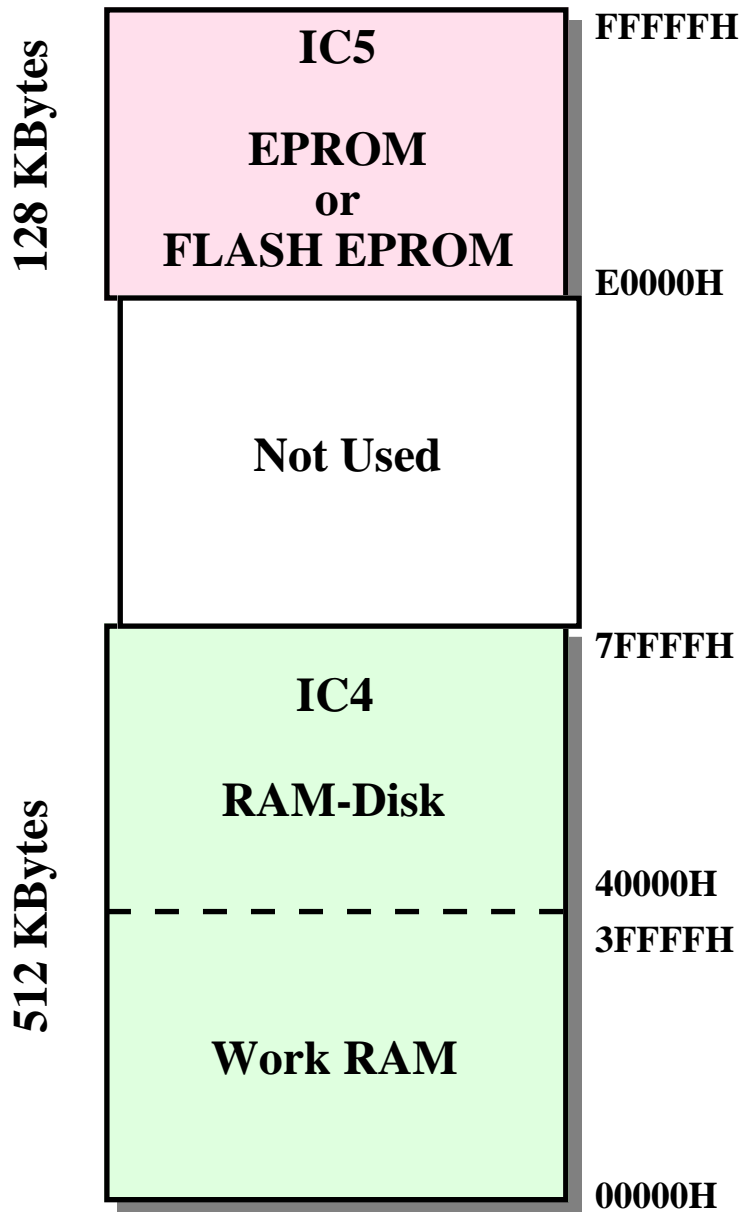
**ORGANIZZAZIONE DELLA MEMORIA SU GPC® 883**

All'atto del reset o power-on, il **GDOS 188** verifica la quantità di SRAM (128 o 512 KBytes), quindi viene organizzata la memoria disponibile come illustrato nelle due figure di seguito.



**FIGURA 13: ORGANIZZAZIONE MEMORIA SU GPC® 883 CON SRAM DA 128 KBYTES**

Come si può notare dalla figura, con questa configurazione di memoria non é disponibile la RAM-Disk, in quanto tutta la SRAM installata sulla **GPC® 883**, viene gestita dal **GDOS 188** come RAM di lavoro.

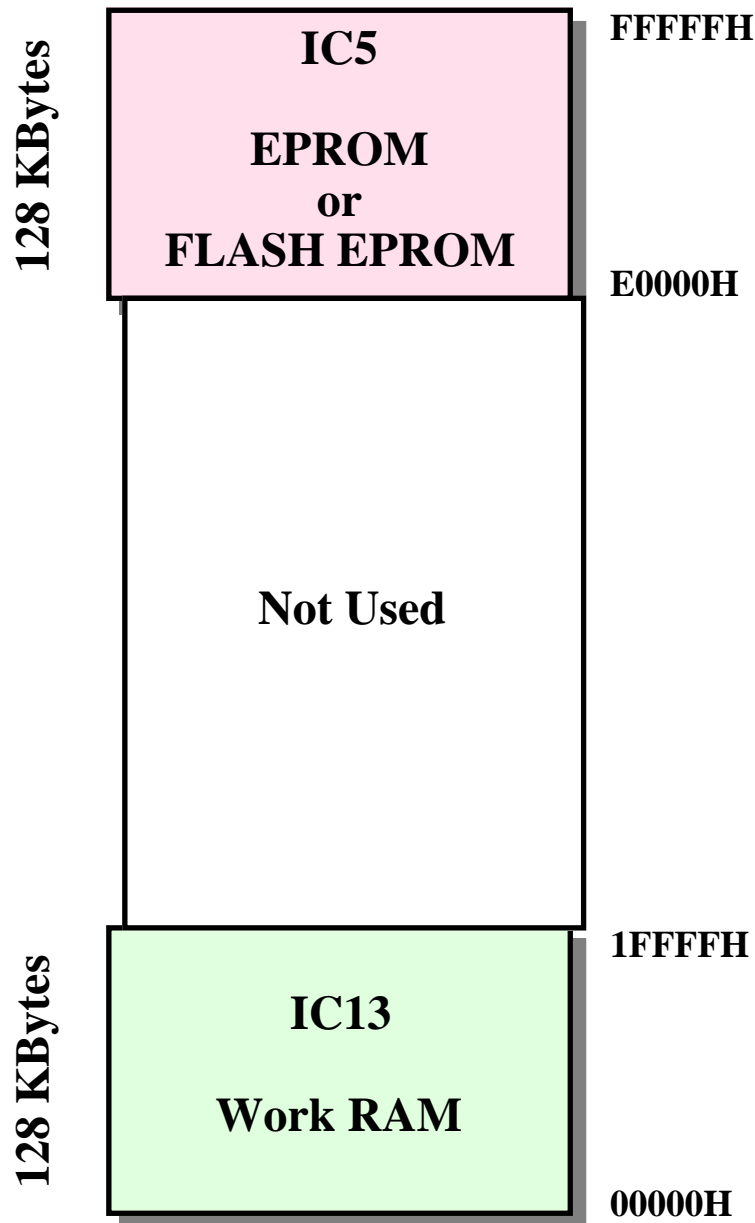


**FIGURA 14: ORGANIZZAZIONE MEMORIA SU GPC® 883 CON SRAM DA 512 KBYTES**

Come si può notare dalla figura, con questa configurazione di memoria, la SRAM installata sulla **GPC® 883**, viene gestita dal **GDOS 188** come RAM di lavoro solo nei primi 256 KBytes, mentre la restante é disponibile per essere utilizzata come RAM-Disk.

**ORGANIZZAZIONE DELLA MEMORIA SU GPC® 884**

All'atto del reset o power-on, il **GDOS 188** verifica la quantità di SRAM (128 o 512 KBytes), quindi viene organizzata la memoria disponibile come illustrato nelle due figure di seguito.



**FIGURA 15: ORGANIZZAZIONE MEMORIA SU GPC® 884 CON SRAM DA 128 KBYTES**

Come si può notare dalla figura, con questa configurazione di memoria non é disponibile la RAM-Disk, in quanto tutta la SRAM installata sulla **GPC® 884**, viene gestita dal **GDOS 188** come RAM di lavoro.

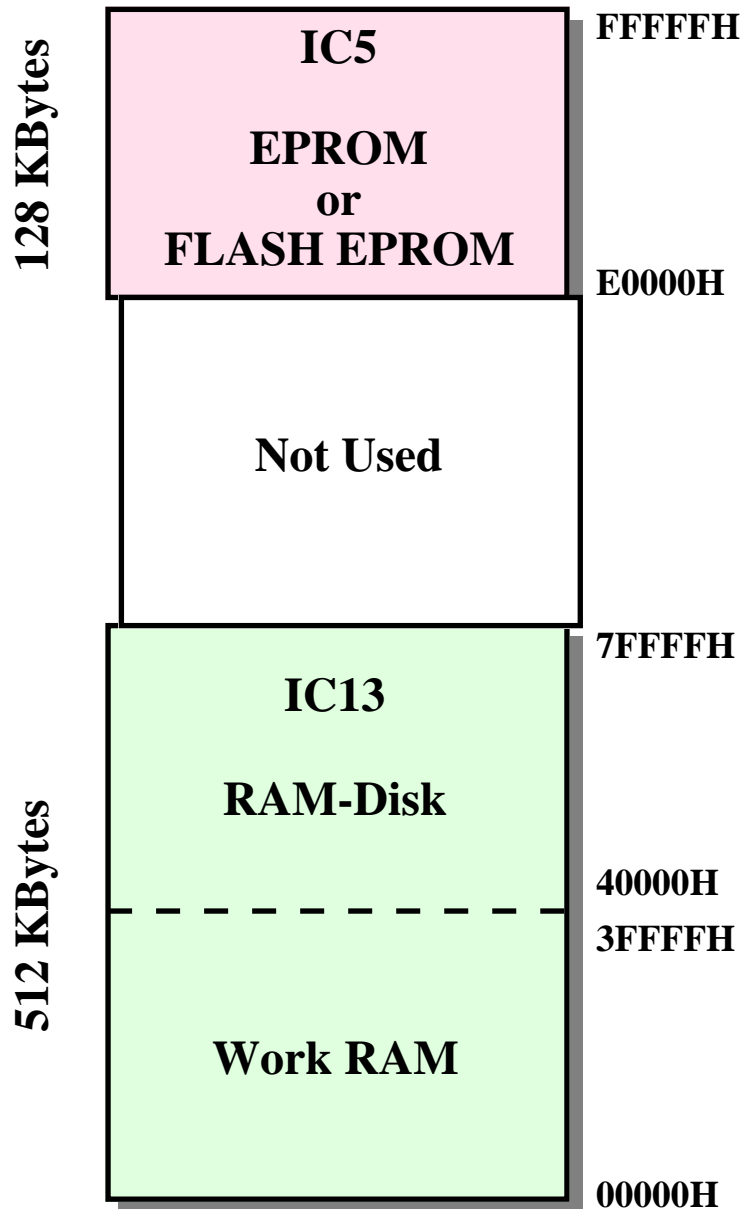


FIGURA 16: ORGANIZZAZIONE MEMORIA SU GPC® 884 CON SRAM DA 512 KBYTES

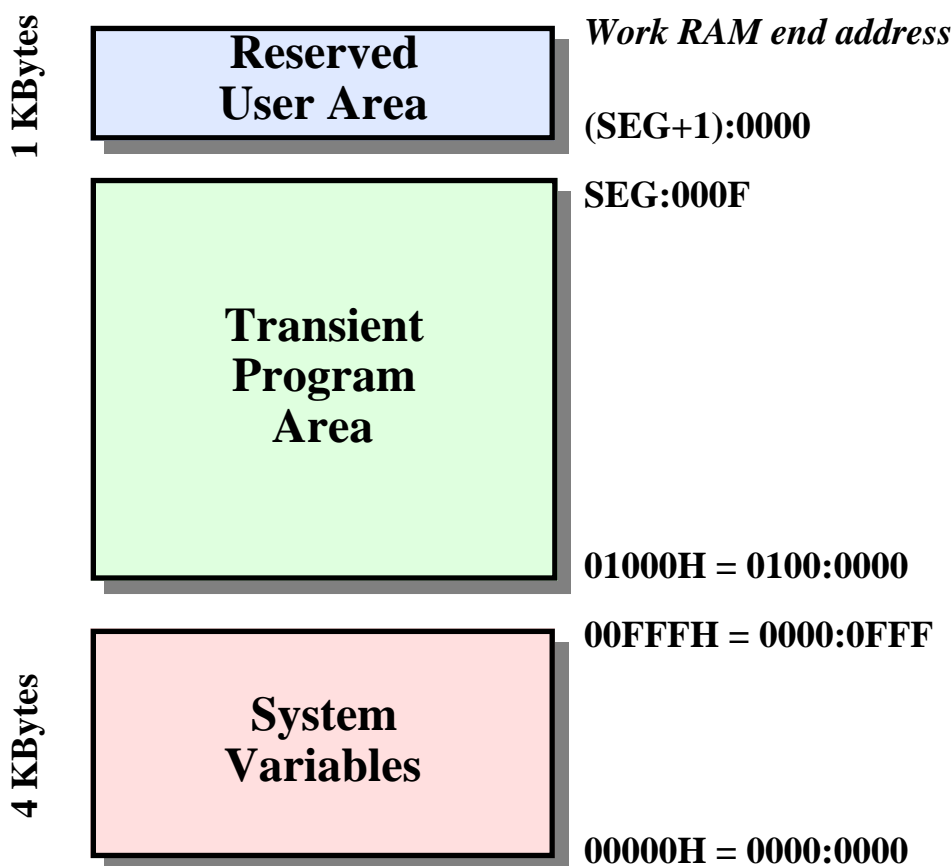
Come si può notare dalla figura, con questa configurazione di memoria, la SRAM installata sulla GPC® 884, viene gestita dal GDOS 188 come RAM di lavoro solo nei primi 256 KBytes, mentre la restante é disponibile per essere utilizzata come RAM-Disk.

**ORGANIZZAZIONE DELLA RAM DI LAVORO**

Il sistema operativo **GDOS 188**, organizza la quantità di RAM di lavoro a sua disposizione, dividendola in tre parti: la prima parte (**4 KBytes**), é utilizzata per le proprie variabili di sistema; la seconda parte di **1 KBytes** é completamente a disposizione utente per l'eventuale salvataggio di dati e parametri; la terza area, rimanente, (denominata **T.P.A.: Transient Program Area**),é invece disponibile per il programma applicativo.

Il compilatore **PASCAL 188**, se usato, utilizza direttamente quest'area di memoria per le sue varie strutture: area dati, area codice, Heap, Stack, ecc.; in questo caso l'utente non deve preoccuparsi di gestire la **TPA**, ma potrà accedere ad essa tramite le strutture dati tipiche del **PASCAL**, quali ad esempio variabili, vettori, puntatori, insiemi, liste, ecc.

L'organizzazione della RAM di lavoro é riportata nella seguente figura:



**FIGURA 17: ORGANIZZAZIONE DELLA RAM DI LAVORO**

L'indirizzo finale dell'area riservata all'utente (**1 KBytes**), denominato *Work RAM end address*, é ricavabile dalle indicazioni riportate nei paragrafi precedenti ed assume i seguenti valori:

- GPC® 188D:** IND1 (**J2** connesso)  
IND3 (**J2** non connesso)
- GPC® 188F:** 1FFFFH = 1F00:0FFF (**J2** connesso)  
3FFFFH = 3F00:0FFF (**J2** non connesso)
- GPC® 883 e GPC® 884:** 1FFFFH = 1F00:0FFF (SRAM = 128 KBytes)  
3FFFFH = 3F00:0FFF (SRAM = 512 KBytes)

Gli indirizzi di fine della **TPA** e d'inizio dell'area riservata all'utente, si ricavano dal valore della variabile **SEG**; questa word, come illustrato in seguito può essere acquisita dall'utente e viene settata dal **GDOS 188** in fase di reset o power-on, dopo aver verificato la quantità di RAM disponibile. L'area di 4 KByte dedicata alle variabili di sistema, contiene tre locazioni in cui vi sono delle informazioni, ricavate dal **GDOS 188**, durante la fase di inizializzazione; queste variabili sono a disposizione dell'utente, che può leggerle, o in determinate condizioni, scriverle. Di seguito ne è riportato l'elenco ed il significato.

**Indirizzo 00400H (0040:0000) - Ultimo segmento di TPA disponibile**

Contiene la word **SEG**, descritta in precedenza, la quale indica l'ultimo segmento della RAM di lavoro, che il **GDOS 188** ha destinato come **TPA**. Se, utilizzando il compilatore PASCAL 188, si vuole acquisire questa variabile, sarà necessario inserire la seguente linea di programma:

```
<variabile> := MemW [$0040:$0000];
```

Volendo poi scrivere un byte, nel primo indirizzo dell'area riservata a utente, sarà sufficiente digitare:

```
Mem [<variabile>+1:$0000] := <byte>;
```

**N.B.** Questa locazione di memoria non deve essere alterata in nessun modo dall'utente.

**Indirizzo 00402H (0040:0002) - Flag di CONSOLE**

Permette di specificare al sistema operativo **GDOS 188**, se la linea di console (seriale A della scheda remota), è utilizzata per comunicare con un dispositivo generico, oppure con l'emulazione terminale intelligente del programma **GET188**, il quale sfrutta un apposito protocollo logico di comunicazione. Nel primo caso tale variabile dovrà contenere la word **55AA Hex**, mentre con qualsiasi altro valore, si avrà una modalità di comunicazione relativa al secondo caso.

Se ad esempio, lavorando con il compilatore PASCAL 188, l'utente desidera utilizzare la linea di console, per comunicare con un terminale, della serie **QTP** del carteggio **grifo®**, come prima riga di programma, sarà necessario inserire:

```
MemW [$0040:$0002] := $55AA;
```

**Indirizzo 00404H (0040:0004) - Valore del registro PBA della CPU**

Contiene il valore di settaggio del registro **Peripheral Base Address**, programmato in fase di inizializzazione dal sistema operativo **GDOS 188**. Tale valore coincide con l'indirizzo di inizio mappaggio, delle periferiche di bordo della scheda remota e sarà maggiormente illustrato nel paragrafo "MAPPAGGIO DELLE PERIFERICHE DELLA SCHEDA REMOTA".

Se, utilizzando il compilatore PASCAL 188, si vuole acquisire questa variabile, sarà necessario inserire la seguente linea di programma:

```
<variabile> := MemW [$0040:$0004];
```

**N.B.** Questa locazione di memoria non deve essere alterata in nessun modo dall'utente.



## ORGANIZZAZIONE DELLA EPROM O FLASH EPROM

Il pacchetto **GDOS 188** può essere acquistato nella versione comprendente una EPROM oppure una FLASH EPROM. Questa deve essere installata sullo zoccolo apposito della scheda remota e contiene il sistema operativo, il compilatore PASCAL 188 e l'utility TOOLS 188.

Questi e tutti gli altri programmi memorizzati in EPROM o FLASH EPROM, vengono eseguiti direttamente da tale dispositivo, lasciando quindi disponibile tutta la RAM di lavoro che, infatti, viene utilizzata unicamente, per la gestione di variabili e strutture dai e, in fase di sviluppo del programma utente, per la memorizzazione dello stesso.

La caratteristica di eseguire il codice in EPROM, un'ulteriore sicurezza intrinseca al sistema che si vuole realizzare, nei confronti di rumori del campo, disturbi ,ecc.

Di seguito viene riportata l'organizzazione di EPROM e FLASH EPROM, nelle varie configurazioni possibili. Da notare che nelle figure successive vengono riportati gli indirizzi di allocazione delle aree di memoria, riferiti allo spazio di indirizzamento del microprocessore; tali indirizzi sono shiftati di E0000 Hex, rispetto a quelli fisici del dispositivo stesso.

### ORGANIZZAZIONE DELLA EPROM

La EPROM contenuta nel pacchetto **GDOS 188** e denominata ERPOM di lavoro, è organizzata come illustrato nella figura seguente.

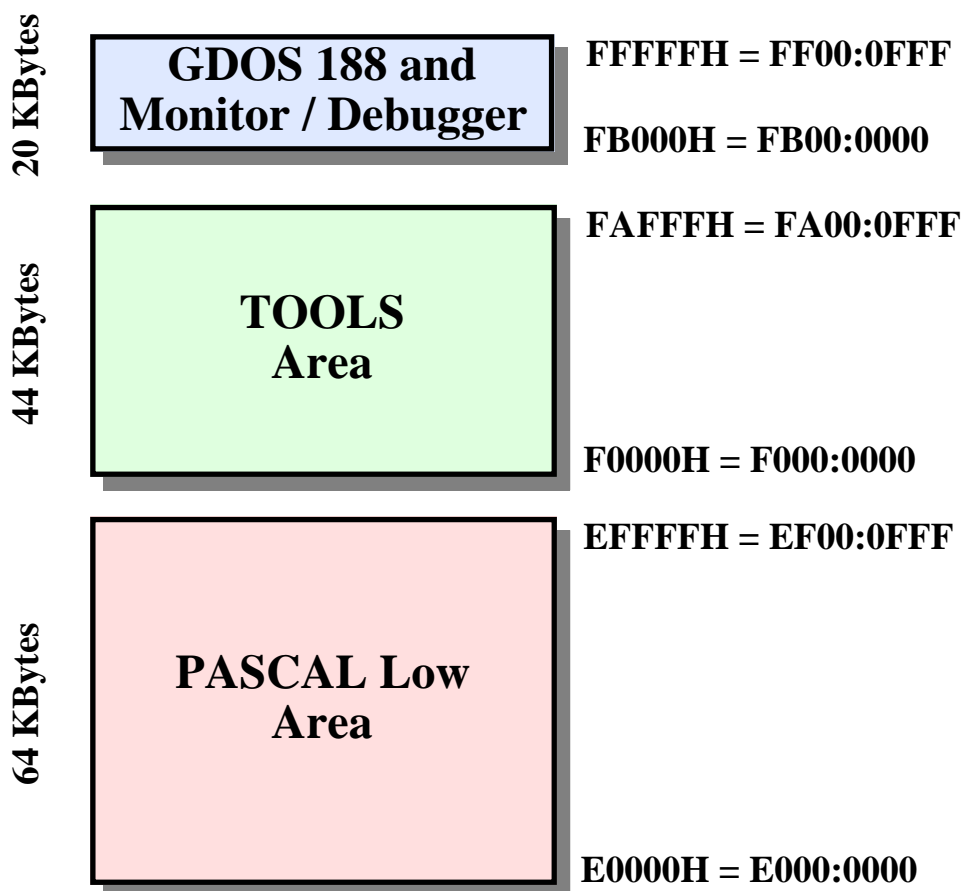
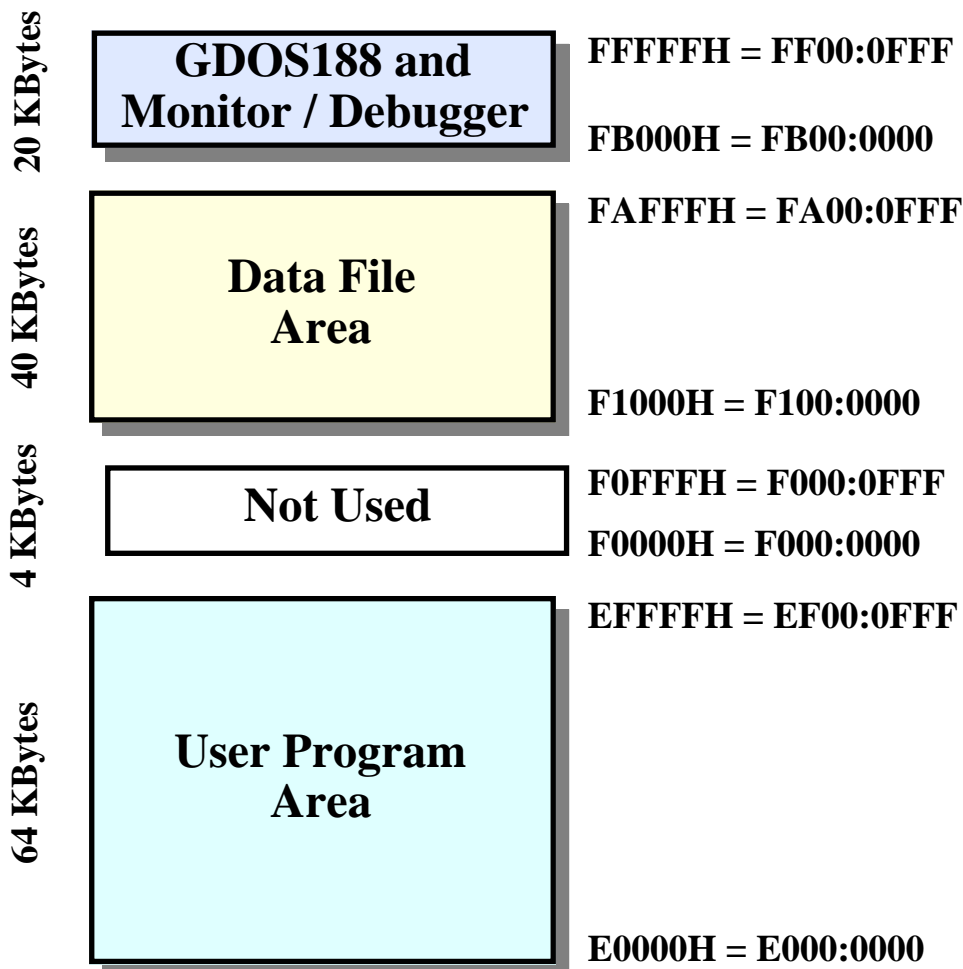


FIGURA 18: ORGANIZZAZIONE DELLA EPROM DI LAVORO

A scopo informativo, viene ora riportata l'organizzazione della EPROM, contenente il programma utente, realizzata tramite l'apposita utility del programma **GET188**.

Infatti tale utility, come illustrato in precedenza, organizza in modo automatico i vari file senza richiedere all'utente, quelli che sono i vari indirizzi di allocazione.



**FIGURA 19: ORGANIZZAZIONE DELLA EPROM UTENTE**

#### *GDOS 188 and Monitor / Debugger*

Occupi gli ultimi 20K Bytes della EPROM e contiene il codice del firmware di base della scheda. Tale codice è quello che viene sempre eseguito al power on o reset della scheda e che poi consente di effettuare tutte le possibili fasi di sviluppo e/o di esecuzione del programma applicativo.

Il contenuto in quest'area corrisponde al file **MON?????.BIN** presente nel disco distribuito.

#### *TOOLS Area*

Area a bassa priorità di AUTORUN, occupa 44 KBytes della EPROM di lavoro e contiene il programma di utility TOOLS 188, corrispondente al file **TOOLS188.COM** del disco di distribuito.

#### *PASCAL Low Area*

Area ad alta priorità di AUTORUN, occupa i primi 64 KBytes della EPROM di lavoro e contiene il compilatore PASCAL 188, corrispondente al file **PAS188\_L.BIN** contenuto del disco distribuito.

#### *User Program Area*

Area ad alta priorità di AUTORUN, occupa i primi 64K Bytes della EPROM utente, contenente il relativo programma di gestione.

*Data File Area*

Occupa 40K Bytes della EPROM utente e può contenere ogni tipo di dato (ad esempio dati di configurazione, messaggi, tabelle, ecc). Tale area sarà gestita direttamente dal programma applicativo, tramite operazioni di lettura in memoria a partire dall'indirizzo **F1000H**.

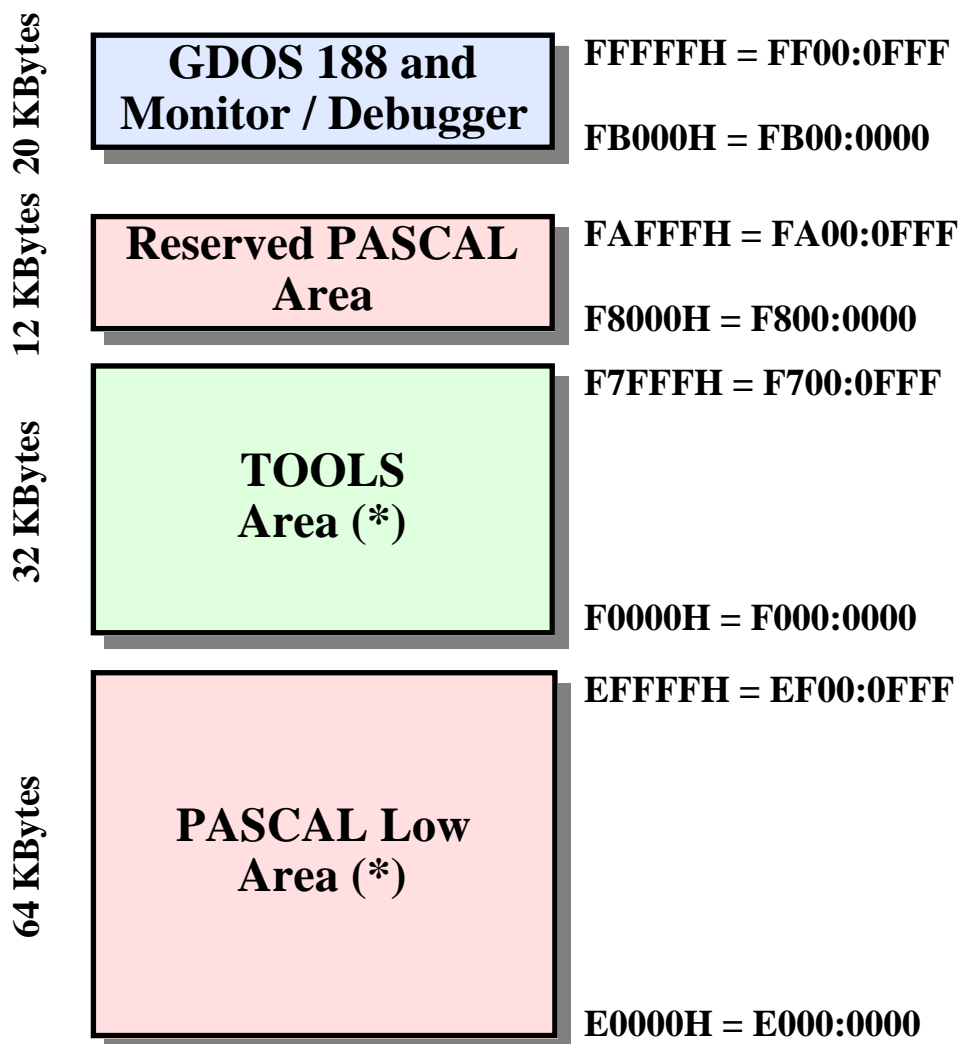
Se, utilizzando il compilatore PASCAL 188, se ne vuole acquisire il primo byte, sarà necessario inserire la seguente riga di programma:

```
<byte> := Mem[$F100 : $0000];
```

Tutti i programmi contenuti nelle aree della EPROM, sono eseguibili dal Monitor/Debugger 188 con appositi comandi (vedere il capitolo "DESCRIZIONE DEL MONITOR / DEBUGGER 188"), però uno solo di questi viene eseguito in modo automatico, secondo le priorità descritte nel paragrafo "PROGRAMMA IN AUTORUN".

**ORGANIZZAZIONE DELLA FLASH EPROM**

La FLASH EPROM contenuta nel pacchetto **GDOS 188** e denominata FLASH EPROM di lavoro, é organizzata come illustrato nella figura seguente.

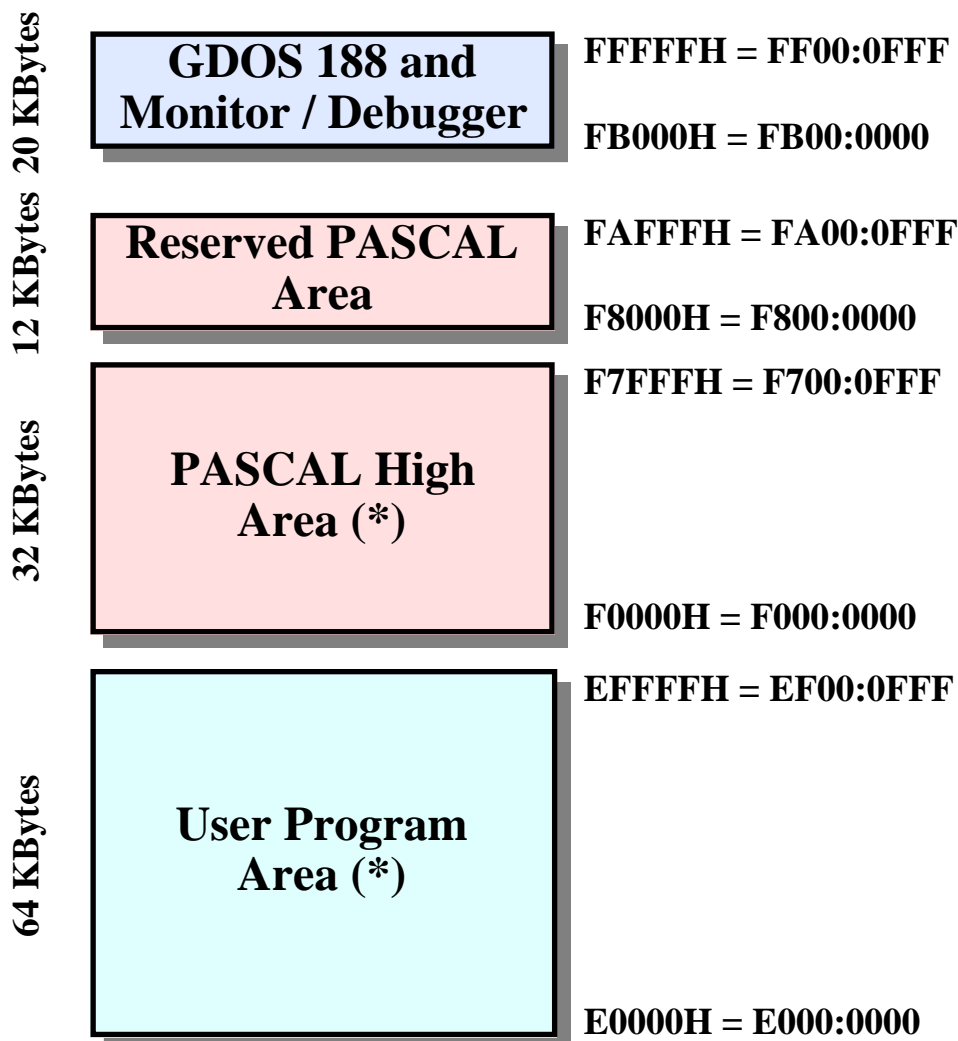


**FIGURA 20: ORGANIZZAZIONE DELLA FLASH EPROM DI LAVORO**

Le aree contrassegnate con (\*) nella figura precedente ed anche in quelle seguenti, possono essere riscritte dal programma di gestione della FLASH EPROM, denominato **WFLASH.HEX**, con le modalità descritte in seguito. La FLASH EPROM di lavoro può essere quindi ricreata in qualsiasi momento utilizzando i file contenuti nel disco di distribuzione.

Le restanti aree occupano interamente gli ultimi due settori da 16 KBytes della FLASH EPROM utilizzabile che, per motivi di sicurezza, non sono mai maneggiati, fornendo quindi una garanzia estrema, sulla validità del codice **GDOS 188** e Monitor/Debugger. Questo va a scapito di una riduzione dell'area per il Data File da 40 a 28K Bytes che comunque è un limite accettabile.

Nelle due figure seguenti, viene riportata l'organizzazione della FLASH EPROM in due possibili configurazioni, ottenibili con il programma WFLASH.HEX; la prima definita "debug" (programma di gestione + compilatore PASCAL 188) e la seconda "utente" (programma di gestione + file dati).



**FIGURA 21: ORGANIZZAZIONE DELLA FLASH EPROM DI DEBUG**

Se non viene utilizzato nessun file di dati, è consigliabile realizzare questa configurazione in quanto la presenza del compilatore PASCAL 188 nella FLASH EPROM, permette di debug-are ed eventualmente modificare il programma utente, senza sostituire il dispositivo installato sulla scheda remota.

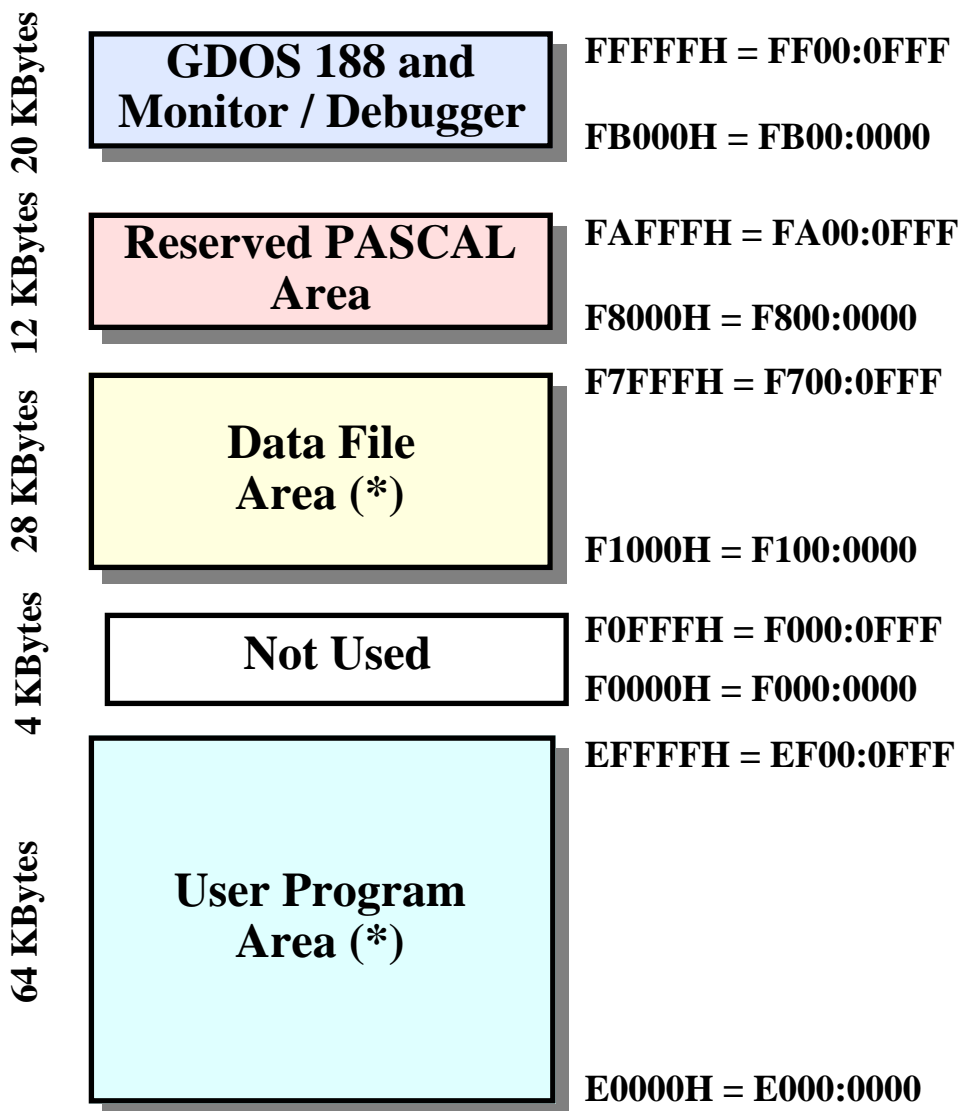


FIGURA 22: ORGANIZZAZIONE DELLA FLASH EPROM UTENTE

*GDOS 188 and Monitor / Debugger*

Occupi gli ultimi 20K Bytes della FLASH EPROM e contiene il codice del firmware di base della scheda. Tale codice è quello che viene sempre eseguito al power on o reset della scheda e che poi consente di effettuare tutte le possibili fasi di sviluppo e/o di esecuzione del programma applicativo. Il contenuto in quest'area corrisponde al file **MON?????.BIN** presente nel disco distribuito e vista la sua importanza non può essere modificato dall'utente ed è inoltre protetto contro ogni variazione accidentale.

*TOOLS Area*

Area a bassa priorità di AUTORUN, occupa 32 KBytes della FLASH EPROM di lavoro e contiene il programma di utility TOOLS 188, corrispondente al file **TOOLS188.COM** del disco distribuito.

*PASCAL Low Area*

Area ad alta priorità di AUTORUN, occupa i primi 64 KBytes della FLASH EPROM di lavoro e contiene il compilatore PASCAL 188, corrispondente al file **PAS188\_L.BIN** contenuto del disco distribuito.

### *Reserved PASCAL Area e PASCAL High Area*

Complessivamente occupano l'area di 44 KBytes della FLASH EPROM, a bassa priorità di AUTORUN, destinata a contenere il compilatore PASCAL 188 nella configurazione di debug.

Questo é stato diviso in due parti, di cui una (memorizzata nella **Reserved PASCAL Area**) é sempre presente nella FLASH EPROM, in quanto appartiene agli ultimi due settori che, come descritto in precedenza non possono essere in alcun modo alterati dall'utente.

La seconda parte invece é contenuta nel file **PAS188\_H.BIN** del disco distribuito e deve essere memorizzata all'interno dell'area di 32 KBytes modificabile dall'utente, denominata **PASCAL High Area**.

### *User Program Area*

Area ad alta priorità di AUTORUN, occupa i primi 64K Bytes della FLASH EPROM utente, contenente il relativo programma di gestione.

### *Data File Area*

Occupa 28K Bytes della FLASH EPROM utente (contro i 40K Bytes su EPROM) e può contenere ogni tipo di dato (ad esempio dati di configurazione, messaggi, tabelle, ecc). Tale area sarà gestita direttamente dal programma applicativo, tramite operazioni di lettura in memoria a partire dall'indirizzo **F1000H**.

Se, utilizzando il compilatore PASCAL 188, se ne vuole acquisire il primo byte, sarà necessario inserire la seguente riga di programma:

```
<byte> := Mem[$F100 : $0000];
```

Tutti i programmi contenuti nelle varie zone della FLASH EPROM, sono eseguibili dal Monitor/Debugger 188 con appositi comandi (vedere il capitolo "DESCRIZIONE DEL MONITOR / DEBUGGER 188"), però uno solo di questi viene eseguito in modo automatico, secondo le priorità descritte nel paragrafo "PROGRAMMA IN AUTORUN".

## GESTIONE DELLA FLASH EPROM

La gestione della FLASH EPROM con il **GDOS 188** è una operazione assistita in cui l'utente tramite un apposito programma (**WFLASH.HEX**, contenuto nel disco distribuito) é in grado di modificare il contenuto di alcune aree della FLASH EPROM, partendo da files salvati su tutti i drives gestiti dal **GDOS 188**. Tutte queste operazioni sono ad alto livello e sono corredate di messaggi di aiuto che assistono l'utente in tutte le fasi.

### CONFIGURAZIONE SCHEDA

Per gestire correttamente la FLASH EPROM, l'utente deve verificare ed eventualmente effettuare le seguenti configurazioni hardware:

- 1) Collegare la linea seriale A (console) della scheda ad una linea seriale del P.C., sfruttando il cavo di comunicazione descritto nel paragrafo "CAVO DI COMUNICAZIONE SERIALE".
- 2) Montare una FLASH EPROM AMD Am29F010, ST M29F010 o equivalente, in cui sia già programmata l'area **GDOS 188** e Monitor/Debugger 188, sullo zoccolo apposito della scheda remota.
- 3) Configurare i jumpers della scheda remota per l'utilizzo di una FLASH EPROM, come descritto nel manuale tecnico della scheda stessa.
- 4) Chiudere il jumper di selezione del modo operativo per attivare la modalità Debug, come incato nel paragrafo "SELEZIONE DELLA MODALITA' RUN / DEBUG".

### CARICAMENTO ED ESECUZIONE PROGRAMMA DI GESTIONE

Per caricare ed eseguire correttamente il programma di gestione della FLASH EPROM sulla scheda utilizzata, i seguenti passi devono essere eseguiti:

- 1) Lanciare il **GET188** sul P.C. di sviluppo collegato alla scheda.
- 2) Selezionare la giusta configurazione della linea seriale del P.C. (Com e Baud Rate), in accordo con l'attuale configurazione hardware della scheda, tramite l'apposita opzione "Options | Serial Port" del **GET188**.
- 3) Attivare l'emulazione terminale selezionando l'opzione "Options | Terminal" del **GET188**.
- 4) Resettare od accendere la scheda remota in modo da eseguire il Monitor/Debugger 188.
- 5) Selezionare l'opzione "Utility | Transmit file to Monitor | HEX Intel File" del **GET188** e quindi selezionare il file **WFLASH.HEX** nella finestra di dialogo che compare. Tale programma é fornito nel disco di distribuzione.
- 6) Attendere la fine della trasmissione del file e quindi premere il tasto <Invio> con cui ricompare il prompt del Monitor/Debugger.
- 7) Fornire il comando GO premendo il tasto "G" seguito dai seguenti indirizzi:

**GO FROM 0100:0000 <Invio> TILL 0100:FFFF <Invio>**

Questi indirizzi sono quelli di default, così se quelli già presentati sono corretti è sufficiente confermarli premendo due volte il tasto <Invio>.

- 8) A questo punto il programma di gestione FLASH EPROM parte, presentando la sua prima pagina di help.

## FASI DEL PROGRAMMA DI GESTIONE

Per gestire correttamente la FLASH EPROM sulla scheda utilizzata, una serie di operazioni devono essere effettuate dall'utente:

- 1) Leggere attentamente le pagine di help, scorrendole con i tasti "N" e "P" e quindi proseguire l'esecuzione premendo il tasto "Invio".
- 2) Selezionare l'operazione richiesta premendo il tasto numerico associato ("0"÷"4"), come presentato nell'apposito menù che compare.

### 3) **Scrittura dell'User Program Area o PASCAL Low Area**

Questa operazione viene selezionata con il tasto "0", quindi viene richiesto di inserire il nome del file da scrivere in quest'area. Tale file deve essere un eseguibile in formato binario (.COM, .BIN) salvato sulla directory corrente del drive attuale. Una volta inserito il file, il programma ne verifica la presenza ed in caso affermativo prosegue, viceversa richiede se si desidera reinserire un nuovo nome file. In caso di file trovato, ne vengono verificate le dimensioni, verificando che non eccedano i **64K Bytes**, viene cancellato il contenuto attuale di tale area e quindi riscritta la stessa, con i dati prelevati dal file selezionato.

La fase di scrittura è illustrata da un apposito messaggio di stato che mostra l'indirizzo attualmente in programmazione; in questa fase l'utente deve solo attendere il completamento dell'operazione e poi verificarne l'esito.

Questa operazione è normalmente utilizzata per cancellare il programma eseguibile ad alta priorità di AUTORUN, e sostituirlo con l'eseguibile del programma applicativo sviluppato.

### 4) **Scrittura della TOOLS Area o PASCAL High Area**

Questa operazione viene selezionata con il tasto "1", quindi viene richiesto di inserire il nome del file da scrivere in quest'area. Tale file deve essere un eseguibile in formato binario (.COM, .BIN) salvato sulla directory corrente del drive attuale. Una volta inserito il file, il programma ne verifica la presenza ed in caso affermativo prosegue, viceversa richiede se si desidera reinserire un nuovo nome file. In caso di file trovato, ne vengono verificate le dimensioni, verificando che non eccedano i **32K Bytes**, viene cancellato il contenuto attuale di tale area e quindi riscritta la stessa, con i dati prelevati dal file selezionato.

La fase di scrittura è illustrata da un apposito messaggio di stato che mostra l'indirizzo attualmente in programmazione; in questa fase l'utente deve solo attendere il completamento dell'operazione e poi verificarne l'esito.

Questa operazione è normalmente utilizzata per inserire il compilatore PASCAL 188, nell'area a bassa priorità di AUTORUN, in modo da poter ottenere una FLASH EPROM in configurazione di Debug.

### 5) **Scrittura della Data File Area**

Questa operazione viene selezionata con il tasto "2", quindi viene richiesto di inserire il nome del file da scrivere in quest'area. Tale file deve essere di tipo qualsiasi, in formato binario, salvato sulla directory corrente del drive attuale. Una volta inserito il file, il programma ne verifica la presenza ed in caso affermativo prosegue, viceversa richiede se si desidera reinserire un nuovo nome file. In caso di file trovato, ne vengono verificate le dimensioni, verificando che non eccedano i **28K Bytes**, viene cancellato il contenuto attuale di tale area e quindi riscritta la stessa, con i dati prelevati dal file selezionato.



La fase di scrittura è illustrata da un apposito messaggio di stato che mostra l'indirizzo attualmente in programmazione; in questa fase l'utente deve solo attendere il completamento dell'operazione e poi verificarne l'esito.

Questa operazione è normalmente utilizzata per aggiungere un'area dati in FLASH EPROM, utilizzata dal programma applicativo sviluppato.

#### 6) Visualizzazione della schermata di aiuto

Questa operazione viene selezionata con il tasto "3" e permette di visualizzare la schermata di help che compare all'avvio del programma.

#### 7) Uscita dal programma

Questa operazione viene selezionata con il tasto "4" e consente di terminare l'esecuzione del programma WFLASH, quindi ritornare al Monitor/Debugger 188.

8) Durante l'esecuzione nella maggioranza delle fasi è sempre possibile interrompere l'esecuzione del programma premendo il tasto "ESC" con cui si ritorna immediatamente in esecuzione del Monitor/Debugger 188.

9) In ogni fase del programma vengono sempre verificati i possibili malfunzionamenti (errore di accesso al file system, errore di cancellazione FLASH EPROM, errore di scrittura FLASH EPROM, ecc.) e nel caso se ne presenti uno, è immediatamente presentato da un apposito messaggio.

10) Tutti i nomi files richiesti dal programma sono nel formato <drive>:<nome>.<estensione> e come drive possono essere utilizzati tutti quelli gestiti dal **GDOS 188**, RAM DISK inclusa.

I tipici esempi d'uso del programma di gestione FLASH EPROM sono:

*Creazione FLASH EPROM di lavoro:*

- 1) Selezionare l'operazione di scrittura della *PASCAL Low Area* e caricare il file **PAS188\_L.BIN**, contenuto nel disco distribuito.
- 2) Selezionare l'operazione di scrittura della *TOOLS Area* e caricare il file **TOOLS188.COM**, contenuto nel disco distribuito.

*Creazione FLASH EPROM di debug:*

- 1) Selezionare l'operazione di scrittura dell'*User Program Area* e specificare il nome del proprio programma applicativo ottenuto dalla compilazione su disco del PASCAL 188.
- 2) Selezionare l'operazione di scrittura della *PASCAL High Area* e caricare il file **PAS188\_H.BIN**, contenuto nel disco distribuito.

*Creazione FLASH EPROM utente:*

- 1) Selezionare l'operazione di scrittura dell'*User Program Area* e specificare il nome del proprio programma applicativo ottenuto dalla compilazione su disco del PASCAL 188.
- 2) Se il programma applicativo prevede un file di dati, selezionare l'operazione di scrittura della *Data File Area* e specificare il nome del proprio file dati.

## GESTIONE DELLA RAM-DISK

Il sistema operativo **GDOS 188**, si preoccupa di gestire la quantità di SRAM eccedente la RAM di lavoro, rendendola interamente disponibile come **RAM-Disk (drive logico M)**.

In questo modo tutti i file gestibili da sistema operativo, possono essere scritti e letti anche sulle risorse locali di memorie di massa.

Questa implementazione consente di accelerare notevolmente tutte quelle che sono le operazioni di caricamento e salvataggio dati, con un grosso risparmio di tempo soprattutto per quanto riguarda lo sviluppo software di tutta l'applicazione.

Usando il compilatore PASCAL, é inoltre possibile utilizzare quest'area di memoria, mediante i comandi ad alto livello tipici della gestione dei file, quali, ad esempio, Assign(<file>,'M:<nome>'), Reset(<file>), Rewrite(<file>), Write(<file>,...), Read(<file>, ....), Close(<file>), ecc.

Il **GDOS 188** in fase di power-on o reset verifica la presenza e la relativa formattazione, della RAM-Disk, in modo da poter sucessivamente gestire questa sezione senza nessuna operazione da parte dell'utente. Nel caso in cui la RAM-Disk sia presente ma non formattata (primo utilizzo o perdita dei dati), l'utente può provvede a farlo utilizzando il comando apposito del programma TOOLS 188, oppure mediante delle chiamate alle varie funzioni del sistema operativo **GDOS 188**. Per ulteriori chiarimenti si consultino i capitoli "DESCRIZIONE DEL PROGRAMMA TOOLS 188" e "FUNZIONI DEL **GDOS 188**".

Vengono di seguito riportati una serie di informazioni che riguardano i limiti di utilizzo della RAM-Disk, quando si lavora con il **GDOS 188**, dovuti alle tecniche di implementazione di tale sistema operativo.

- 1) I nomi dei files che contengono wild card (\* o ?) non sono gestite sulla RAM-Disk e quindi non possono essere utilizzate.
- 2) A seguito della formattazione della RAM-Disk, lo spazio effettivo disponibile risulta essere pari al size desiderato, diminuito di 1 Kbyte, relativo a informazioni sullo stato del disco.
- 3) Lo spazio minimo occupato da un file in RAM-Disk é di **1 KBytes**, anche se questo contiene un numero inferiore di dati.
- 4) Se viene a mancare la tensione di alimentazione della scheda remota, e vi sono uno o più file aperti in RAM-Disk, si verificherà un danneggiamento di quest'ultima, rendendo necessario una nuova formattazione. Questo problema può essere aggirato, aprendo il file in RAM-Disk, solo nel momento in cui si vogliono effettuare delle operazione di lettura o scrittura, e chiudendo lo stesso subito dopo.

Per esempi relativi all'utilizzo della RAM-Disk, fare riferimento agli appositi programmi dimostrativi contenuti nel disco distribuito insieme al pacchetto **GDOS 188**.

## PROGRAMMA IN AUTORUN

Con il sistema operativo **GDOS 188**, 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.

Un programma viene eseguito nella modalità di **AUTORUN**, se contiene una determinata sequenza di codici, illustrata in seguito; inoltre deve essere memorizzato a partire da alcuni indirizzi specifici, oppure deve essere salvato, nella RAM-Disk locale, con il nome: **AUTORUN.COM**.

Il sistema operativo **GDOS 188**, in fase di reset o power-on, se configurato nella modalità **RUN** come descritto in precedenza, verifica la presenza del programma di **AUTORUN**, seguendo una determinata scaletta di priorità, in particolare, vengono fatte le seguenti verifiche:

### 1) *Presenza in RAM-Disk del file **AUTORUN.COM***

Il file con questo nome viene eseguito se contiene la sequenza di **AUTORUN** e se non è frammentato; l'utente, per essere certo di quest'ultima condizione, deve verificare che, in precedenza, non siano stati cancellati dei file, oppure riformattare la RAM-Disk prima del salvataggio del programma **AUTORUN.COM**.

### 2) *Presenza di un programma nella zona ad alta priorità di **AUTORUN***

Il programma memorizzato in EPROM o FLASH EPROM a partire dall'indirizzo **E0000H=E000:0000**, viene eseguito se contiene la sequenza di **AUTORUN**.

### 3) *Presenza di un programma nella zona a bassa priorità di **AUTORUN***

Il programma memorizzato in EPROM o FLASH EPROM a partire dall'indirizzo **F0000H=F000:0000**, viene eseguito se contiene la sequenza di **AUTORUN**.

### 4) *Presenza di un programma a partire dall'indirizzo **01000H = 0100:0000***

Il programma memorizzato nella **TPA** a partire da questo indirizzo, viene eseguito se contiene la sequenza di **AUTORUN**.

La sequenza di **AUTORUN** viene inserita automaticamente all'inizio del programma, nel caso questo sia sviluppato con il compilatore **PASCAL 188**; altrimenti, se l'utente, vuole sviluppare l'applicativo in linguaggio Assembly, come prime righe di programma, deve inserire le seguenti linee:

```

xx00 : 0000           Jmp START
xx00 : 0004           DB Not ("A")           ; Autorun FLAG
xx00 : 0005           DB "A"
xx00 : 0006   START: <Applicativo>

```

Dove le indicazioni **xx00**, si riferiscono agli indirizzi sopra riportati.

Se nessuna delle condizioni descritte è verificata, il sistema operativo **GDOS 188**, esegue il Monitor/Debugger 188.

### **N.B.**

Facendo riferimento alle figure precedenti, relative all'organizzazione di EPROM o FLASH EPROM, si può notare che all'atto del reset o power-on, il sistema operativo **GDOS 188**, se non trova il programma **AUTORUN.COM** in RAM-Disk, esegue il compilatore **PASCAL 188** (struttura di lavoro) oppure il programma utente (struttura utente o di debug).

## MAPPAGGIO DELLE PERIFERICHE

Il sistema operativo **GDOS 188**, all'atto del del reset o del power-on, programma il registro interno della CPU, denominato **PBA (Peripheral Base Address)**, con il valore **F000H**. In questo modo vengono configurati i vari chip select del microprocessore e le relative periferiche della scheda remota, ad essi associate (come indicato nel MANUALE TECNICO della scheda).

Nei paragrafi seguenti viene riportato l'elenco completo di tali indirizzi di mappaggio, in funzione delle varie schede **GPC®** utilizzabili dal **GDOS 188**.

### **MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188D**

L'elenco completo degli indirizzi di mappaggio delle periferiche sulla scheda **GPC® 188D** è riportato nelle due tabelle seguenti.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>W.DOG</b>	RWD	F000H	R	Retrigger watch dog
<b>EEPROM</b>	RE2	F000H	R/W	Accesso seriale ad EEPROM
<b>SCC 85C30</b>	RSB	F080H	R/W	Registro stato linea seriale B
	RDB	F081H	R/W	Registro dati linea seriale B
	RSA	F082H	R/W	Registro stato linea seriale A
	RDA	F083H	R/W	Registro dati linea seriale A
<b>DMA</b>	DMA	F100H	R/W	Registro disattivazione richiesta DMA
<b>ABACO® I/O BUS</b>	IOBUS	F180H÷F1FFH	R/W	Indirizzi per la gestione dell' <b>ABACO®</b> I/O BUS
<b>PPI 82C55</b>	PA	F200H	R/W	Registro dati del port A
	PB	F201H	R/W	Registro dati del port B
	PC	F202H	R/W	Registro dati del port C
	RC	F203H	R/W	Registro di controllo e comando

**FIGURA 23: TABELLA 1 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188D**

## NOTA BENE

Come si può notare dalla tabella, la sezione di **ABACO® I/O BUS** della **GPC®188D** è mappata in **128** indirizzi, compresi al range **F180H÷F1FFH** (indirizzi con **A7** a livello logico 1).

Per questo motivo, se una scheda periferica collegata a tale sezione, fa uso di tutte le 8 linee di indirizzamento **A0÷A7**, questa dovrà essere mappata a partire da un indirizzo compreso nel campo **128÷255 (80H÷FFH)**.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>RTC 62421</b>	S1	F280H	R/W	Registro unità secondi
	S10	F281H	R/W	Registro decine secondi
	MI1	F282H	R/W	Registro unità minuti
	MI10	F283H	R/W	Registro decine minuti
	H1	F284H	R/W	Registro unità ore
	H10	F285H	R/W	Registro decine ore; AM/PM
	D1	F286H	R/W	Registro unità giorno
	D10	F287H	R/W	Registro decine giorno
	MO1	F288H	R/W	Registro unità mese
	MO10	F289H	R/W	Registro decine mese
	Y1	F28AH	R/W	Registro unità anno
	Y10	F28BH	R/W	Registro decine anno
	W	F28CH	R/W	Registro giorno della settimana
	REGD	F28DH	R/W	Registro di stato e controllo D
	REGE	F28EH	R/W	Registro di stato e controllo E
REGF	F28FH	R/W	Registro di stato e controllo F	
<b>WRITE PROTECT</b>	WRP	F300H	W	Registro sprotezione scrittura SRAM IC9
<b>DIP SWITCH</b>	DSW1	F300H	R	Registro acquisizione dip switch
<b>LD3,4</b>	LED	F340H	W	Registro di gestione LEDS attività

**FIGURA 24: TABELLA 2 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188D**

Per ulteriori chiarimenti e per la descrizione dei registri sopra riportati, si consulti il MANUALE TECNICO della scheda **GPC® 188D**.

**MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188F**

L'elenco completo degli indirizzi di mappaggio delle periferiche sulla scheda **GPC® 188F** é riportato nelle due tabelle seguenti.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>W.DOG</b>	RWD	F000H	R	Retrigger watch dog
<b>EEPROM</b>	RE2	F000H	R/W	Accesso seriale ad EEPROM
<b>MMU</b>	MMU	F000H	R/W	Impaginazione memorie con MMU
<b>LD3,4</b>	LED	F000H	R/W	Registro gestione LEDs di attività
<b>BT1</b>	BAT	F000H	R	Registro di acquisizione stato batteria
<b>SCC 85C30</b>	RSB	F080H	R/W	Registro stato linea seriale B
	RDB	F081H	R/W	Registro dati linea seriale B
	RSA	F082H	R/W	Registro stato linea seriale A
	RDA	F083H	R/W	Registro dati linea seriale A
<b>DMA</b>	DMA	F100H	R/W	Registro disattivazione richiesta DMA
<b>A/D LM12458</b>	IRL0÷7	F180H÷F18EH (Ind. PARI)	R/W	Registro istruzioni low 0÷7 del sequencer
	IRH0÷7	F181H÷F18FH (Ind. DISPARI)	R/W	Registro istruzioni high 0÷7 del sequencer
	CNTL	F190H	R/W	Registro di configurazione low
	CNTH	F191H	R/W	Registro di configurazione high
	INTENL	F192H	R/W	Registro abilitazione interrupt low
	INTENH	F193H	R/W	Registro abilitazione interrupt high
	INTSTL	F194H	R	Registro di stato interrupt low
	INTSTH	F195H	R	Registro di stato interrupt high
	TMRL	F196H	R/W	Registro per timer low
	TMRH	F197H	R/W	Registro per timer high
	FIFOL	F198H	R	Registro per conversioni in FIFO low
	FIFOH	F199H	R	Registro per conversioni in FIFO high
	LIMSTL	F19AH	R	Registro stato limiti low
	LIMSTH	F19BH	R	Registro stato limiti high

**FIGURA 25: TABELLA 1 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188F**

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>PPI 82C55</b>	PA	F200H	R/W	Registro dati del port A
	PB	F201H	R/W	Registro dati del port B
	PC	F202H	R/W	Registro dati del port C
	RC	F203H	R/W	Registro di controllo e comando
<b>RTC 62421</b>	S1	F280H	R/W	Registro unità secondi
	S10	F281H	R/W	Registro decine secondi
	MI1	F282H	R/W	Registro unità minuti
	MI10	F283H	R/W	Registro decine minuti
	H1	F284H	R/W	Registro unità ore
	H10	F285H	R/W	Registro decine ore; AM/PM
	D1	F286H	R/W	Registro unità giorno
	D10	F287H	R/W	Registro decine giorno
	MO1	F288H	R/W	Registro unità mese
	MO10	F289H	R/W	Registro decine mese
	Y1	F28AH	R/W	Registro unità anno
	Y10	F28BH	R/W	Registro decine anno
	W	F28CH	R/W	Registro giorno della settimana
	REGD	F28DH	R/W	Registro di stato e controllo D
	REGE	F28EH	R/W	Registro di stato e controllo E
	REGF	F28FH	R/W	Registro di stato e controllo F
<b>WRITE PROTECT</b>	WRP	F300H	W	Registro sprotezione scrittura SRAM IC9
<b>DIP SWITCH</b>	DSW1	F300H	R	Registro acquisizione dip switch

**FIGURA 26: TABELLA 2 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 188F**

Per ulteriori chiarimenti e per la descrizione dei registri sopra riportati, si consulti il MANUALE TECNICO della scheda **GPC® 188F**.

**MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 883**

L'elenco completo degli indirizzi di mappaggio delle periferiche sulla scheda **GPC® 883** é riportato nelle due tabelle seguenti.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>ABACO® I/O BUS</b>	I/O BUS	F000H÷F0FFH	R/W	Indirizzi gestione <b>ABACO®</b> I/O BUS
<b>RUN / DEB.</b>	RUNDEB	F100H	R	Registro acquisizione stato jumper J1
<b>RTC 62421</b>	SEC1	F100H	R/W	Registro dati per unità secondi
	SEC10	F101H	R/W	Registro dati per decine secondi
	MIN1	F102H	R/W	Registro dati per unità minuti
	MIN10	F103H	R/W	Registro dati per decine minuti
	HOU1	F104H	R/W	Registro dati per unità ore
	HOU10	F105H	R/W	Registro dati per decine ore e AM/PM
	DAY1	F106H	R/W	Registro dati per unità giorno
	DAY10	F107H	R/W	Registro dati per decine giorno
	MON1	F108H	R/W	Registro dati per unità mese
	MON10	F109H	R/W	Registro dati per decine mese
	YEA1	F10AH	R/W	Registro dati per unità anno
	YEA10	F10BH	R/W	Registro dati per decine anno
	WEE	F10CH	R/W	Registro dati per giorno della settimana
	REGD	F10DH	R/W	Registro di controllo D
	REGE	F10EH	R/W	Registro di controllo E
REGF	F10FH	R/W	Registro di controllo F	
<b>A/D MAX197</b>	ADCTRL	F500H	W	Registro di controllo A/D converter
	ADL	F501H	R	Registro dati low A/D converter
	ADH	F502H	R	Registro dati high A/D converter
<b>PPI 82C55</b>	PA	F540H	R/W	Registro dati del port A
	PB	F541H	R/W	Registro dati del port B
	PC	F542H	R/W	Registro dati del port C
	RC	F543H	R/W	Registro di controllo e comando

**FIGURA 27: TABELLA 1 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 883**



DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>W. DOG</b>	RWD	F580H	R/W	Registro di retrigger del Watch Dog
<b>DIP SWITCH</b>	DSW1	F5C0H	R	Registro di acquisizione dip switch
<b>UART CAN SJA 1000</b>	CAN	F600H+F67FH	R/W	Registri per la gestione dell'UART CAN SJA 1000, in modalità BasicCAN o PeliCAN (i registri sono gli stessi riportati nel data sheet del componente, con un <u>offset</u> di <b>F600H</b> ).

**FIGURA 28: TABELLA 2 DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 883**

Per ulteriori chiarimenti e per la descrizione dei registri sopra riportati, si consulti il MANUALE TECNICO della scheda **GPC® 883**.

**MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 884**

L'elenco completo degli indirizzi di mappaggio delle periferiche sulla scheda **GPC® 884** é riportato nella tabella seguente.

DISP.	REG.	INDIRIZZO	R/W	SIGNIFICATO
<b>ABACO® I/O BUS</b>	I/O BUS	F000H÷F0FFH	R/W	Indirizzi per la gestione dell' <b>ABACO®</b> I/O BUS
<b>RUN / DEB.</b>	RUNDEB	F100H	R	Registro acquisizione stato jumper J1
<b>RTC 62421</b>	SEC1	F100H	R/W	Registro dati per unità secondi
	SEC10	F101H	R/W	Registro dati per decine secondi
	MIN1	F102H	R/W	Registro dati per unità minuti
	MIN10	F103H	R/W	Registro dati per decine minuti
	HOU1	F104H	R/W	Registro dati per unità ore
	HOU10	F105H	R/W	Registro dati per decine ore e AM/PM
	DAY1	F106H	R/W	Registro dati per unità giorno
	DAY10	F107H	R/W	Registro dati per decine giorno
	MON1	F108H	R/W	Registro dati per unità mese
	MON10	F109H	R/W	Registro dati per decine mese
	YEA1	F10AH	R/W	Registro dati per unità anno
	YEA10	F10BH	R/W	Registro dati per decine anno
	WEE	F10CH	R/W	Registro dati per giorno della settimana
	REGD	F10DH	R/W	Registro di controllo D
	REGE	F10EH	R/W	Registro di controllo E
REGF	F10FH	R/W	Registro di controllo F	
<b>W. DOG</b>	RWD	F600H	R/W	Registro di retrigger del Watch Dog

**FIGURA 29: TABELLA DI MAPPAGGIO DELLE PERIFERICHE DELLA GPC® 884**

Per ulteriori chiarimenti e per la descrizione dei registri sopra riportati, si consulti il **MANUALE TECNICO** della scheda **GPC® 884**.

## DESCRIZIONE DEL MONITOR / DEBUGGER 188

Il pacchetto **GDOS 188**, comprende un potente e versatile **Monitor/Debugger**. Questo software consente di effettuare delle operazioni, che riguardano le risorse hardware di base della scheda remota e di debuggare, a livello Assembly, un programma utente. Alcune delle operazioni possibili sono input/output, verifica o sostituzione del contenuto della memoria, gestione registri della CPU, esecuzione passo-passo di un applicativo, settaggio break-point, ecc.

All'atto del reset o del power-on, il sistema operativo **GDOS 188**, verifica lo stato del jumper di selezione del modo operativo (descritto nel precedente capitolo) e controlla la presenza del programma di AUTORUN; se tale file non é presente, oppure é selezionata la modalità **Debug**, viene eseguito il Monitor/Debugger 188 e vengono visualizzate le seguenti informazioni:

MONITOR 188 - **GPC°** ???? ff MHz Vx.x

<Informazioni su alcuni registri>

M188>

Dove con la indicazione generica composta da x, si intende il numero di versione, con ???? la scheda remota utilizzata e con ff la frequenza della stessa.

Nei paragrafi che seguono, vengono riportate le informazioni relative al formato dei parametri del Monitor/Debugger 188, e la serie di comandi che questo supporta.

### FORMATO DEI PARAMETRI

Alcuni comandi del Monitor/Debugger 188, richiedono uno o più parametri, che l'utente deve inserire seguendo un formato opportuno. Questi possono essere di tre tipi diversi, in particolare possono essere richiesti come:

#### *INDIRIZZO ESTESO - ssss:0000*

E' richiesto l'inserimento sia dell'indirizzo di SEGMENTO, indicato ssss, che quello di OFFSET, corrispondente all'indicazione 0000.

#### *OFFSET - 0000 oppure SSSS:0000*

E' richiesto l'inserimento del solo indirizzo di OFFSET, indicato 0000; infatti anche se viene visualizzato anche l'indirizzo di segmento SSSS, questo non potrà essere variato.

#### *BYTE oppure WORD - bb oppure wwww*

E' richiesto l'inserimento di una variabile, sotto forma di byte, indicato bb, oppure di word, corrispondente all'indicazione wwww, nel caso di comandi quali ad esempio output di un dato, scrittura di Registri, fill di un'area di memoria, ecc.

L'inserimento dei parametri sopra descritto, viene agevolato presentando subito il valore di default o quello precedente, relativamente al comando che si va ad utilizzare; tale indicazione può essere confermata con il tasto <Invio>, variata inserendo direttamente il nuovo valore, oppure modificata con il tasto <Back Space>. Nel caso di indirizzo esteso, é inoltre possibile cancellare totalmente tale parametro premento il tasto <Spazio>.

## VETTORI DI INTERRUPT

Il Monitor/Debugger 188, per l'inserimento dei Break-Point, necessari alle varie operazioni di Debug, utilizza i vettori di Interrupt 2 e 3. Questi non devono essere utilizzati nel programma utente.

## COMANDI DEL MONITOR / DEBUGGER 188

Il Monitor/Debugger 188 dispone di una serie di comandi, che permettono di effettuare tutte le funzioni tipiche dei programmi di questo genere; tali istruzioni vengono invocate premendo un singolo tasto, quando è visualizzato il carattere di prompt: **M188>** .

A questo punto il Monitor/Debugger 188 risponde con l'eco del comando ed eventualmente, attende l'inserimento dei parametri necessari; completata questa fase, viene eseguita l'istruzione richiesta, con la visualizzazione dei relativi esiti, quindi viene ripresentato il prompt.

Ogni comando può essere abortito in qualsiasi momento con il tasto <Esc>.

Di seguito viene riportato l'elenco completo delle istruzioni, il tasto ed i parametri relativi, seguiti da un breve descrizione del comando stesso.

## ASSEMBLATORE IN LINEA

*Sintassi:*            A    [*Inizio = INDIRIZZO ESTESO*]            <Invio>

Assembla i codici mnemonici del linguaggio assembler per **8086**, direttamente in memoria, a partire dall'indirizzo specificato. Per uscire dall'assemblatore in linea e ritornare al prompt del Monitor/Debugger premere il tasto <Esc>.

**Utilizzo dei codici mnemonici**      Il codice mnemonico per un ritorno di tipo FAR è **retf**. I codici mnemonici per la manipolazione delle stringhe devono specificare in maniera esplicita la dimensione della stringa. Ad esempio, per spostare stringhe di word (16 bit), va usato **movsw**; per spostare stringhe di byte (8 bit), va usato **movsb**.

**Assemblaggio di salti e chiamate**    L'assembler assembla automaticamente le chiamate ed i salti SHORT, NEAR o FAR, a seconda della disposizione dei byte nell'indirizzo destinazione. E' possibile non tener conto di questi salti e di queste chiamate tramite il prefisso **near** (che si può abbreviare con l'indicazione **ne**) o **far**, come nell'esempio che segue:

```
M188> A 0100:0500
A>0100:0500  jmp  502      ; Un salto short di 2 byte
A>0100:0500  jmp  near 505    ; Un salto near di 3 byte
A>0100:0500  jmp  far  50A   ; Un salto far di 5 byte
```

**Distinzione tra locazioni di memoria di byte e word**    Quando un operando può riferirsi sia ad una locazione di memoria di word sia ad una locazione di memoria di byte, è necessario specificare il tipo di dati dell'operando utilizzando i prefissi **word ptr** oppure **byte ptr**. Abbreviazioni valide sono **wo** e **by**. Ad esempio:

```
dec  wo      [si]
neg  byte ptr [128]
```

**Come specificare gli operandi** L'assemblatore in linea utilizza la convenzione standard secondo cui gli operandi racchiusi tra parentesi quadre si riferiscono ad una locazione di memoria. Ciò accade perché l'assemblatore non è in grado di distinguere tra un operando immediato ed un operando che indica una locazione di memoria. L'esempio seguente chiarisce la differenza:

```
mov    ax, 21    ; Copia il valore 21h nel registro AX
mov    ax, [21]  ; Copia nel registro AX il contenuto della locazione
                    ; di memoria 21h
```

**Utilizzo delle pseudoistruzioni** L'assemblatore in linea dispone di due pseudoistruzioni molto comuni: il codice operativo **db**, che assembla i valori di byte direttamente in memoria ed il codice operativo **dw**, che assembla i valori di word. Seguono degli esempi di entrambi i codici:

```
db     1, 2, 3, 4, "QUESTO E' UN ESEMPIO"
db     'QUESTE SONO VIRGOLETTE: " "'
dw     1000, 2000, 3000, "BACH"
```

L'assemblatore in linea riconosce, inoltre, tutte le forme di comandi indiretti di registro. Ad esempio:

```
add    bx, 34 [bp+2] . [si-1]
pop    [bp+di]
push   [si]
```

## ESECUZIONE PROGRAMMA ALL'INDIRIZZO E0000H

*Sintassi:*            **B**

Manda in esecuzione il programma memorizzato in EPROM o FLASH EPROM, nella zona ad alta priorità di AUTORUN, corrispondente all'indirizzo **E0000H (E000:0000)**, senza controllarne sia la presenza che la sequenza di AUTORUN.

Nel caso di EPROM o FLASH EPROM con il pacchetto software **GDOS 188**, viene eseguito il compilatore PASCAL 188 che, come indicato in precedenza, è memorizzato a partire da tale locazione.

## COPIA DI UN BLOCCO DI MEMORIA

*Sintassi:*            **C**    **[Inizio Blocco = INDIRIZZO ESTESO]**    **<Invio>**  
    **[Fine Blocco = OFFSET]**                            **<Invio>**  
    **[Destinazione = INDIRIZZO ESTESO]**    **<Invio>**

Copia il blocco di memoria, determinato dagli indirizzi "Inizio Blocco" e "Fine Blocco", a partire dalla locazione, della RAM di lavoro, specificata tramite l'indirizzo "Destinazione"

**Esempio** Se si vuole copiare il blocco di memoria di 64 Kbyte, della EPROM o FLASH EPROM, compreso negli indirizzi E0000H÷EFFFFH (da E000:0000 a E000:FFFF), a partire dalla locazione 0100H (0100:0000), della RAM di lavoro, sarà necessario inserire:

```
M188> COPY FROM E000:0000<Invio> TILL E000:FFFF<Invio> TO 0100:0000<Invio>
```









## ESECUZIONE DI UNA PROCEDURA

*Sintassi:*            **P**                            <Invio>

Esegue un passo di programma a partire dall'indirizzo esteso, determinato dai registri **CS:IP**; se lo step in questione, contiene un'istruzione di CALL, la procedura invocata viene eseguita per intero; viceversa viene eseguita la singola istruzione. Al termine di tale comando vengono visualizzati tutti i vari registri ed il contenuto, deassemblato, della successiva istruzione del programma.

**NOTA BENE**    Tale comando comporta l'inserimento di un Break-Point nell'applicativo che si vuole eseguire, quindi questo programma deve necessariamente risiedere nella RAM di lavoro.

## RIAVVIA IL GDOS 188

*Sintassi:*            **Q**

Riavvia completamente il sistema operativo **GDOS 188**, controllando nuovamente lo stato dei jumper e la presenza del programma di AUTORUN, come se si fosse verificata condizione di reset o di power-on di tutto il sistema.

## VISUALIZZAZIONE O MODIFICA DEI REGISTRI

*Sintassi:*            **R**    [*Nome Registro (opzionale)*]            <Invio>

Visualizza o modifica il contenuto dei registri della CPU **80c188** o sua derivata.

Se viene omissso il parametro "Nome Registro" sarà visualizzato il contenuto di tutti i registri, in caso contrario potrà essere alterata la word contenuta nel registro specificato.

**Esempio**    Se vi vuole memorizzare la word 55AAH, all'interno del registro AX, sarà necessario inserire:

M188> REGISTER name (<Enter> show all registers): **AX** <Invio> = **55AA** <Invio>

## SOSTITUZIONE DI DATI IN MEMORIA

*Sintassi:*            **S**    [*Inizio = INDIRIZZO ESTESO*] <Invio>

Permette di sostituzione una serie di byte, nella RAM di lavoro, a partire dall'indirizzo specificato tramite il parametro "Inizio". Una volta confermato tale comando, verranno visualizzate le seguenti informazioni:

S> **ssss:0000** = **bb** WITH

Dove, **ssss:0000** é la locazione iniziale, precedentemente inserita, e **bb** é il dato attualmente contenuto in essa.

A questo punto sono possibile le seguenti operazioni:

< <i>tasto</i> ->	Per visualizzare la locazione di memoria precedente.
< <i>Invio</i> >	Per visualizzare la locazione di memoria successiva.
[ <i>Nuovo Dato = BYTE</i> ] < <i>tasto</i> ->	Sostituisce il dato, nella locazione attuale, con il nuovo byte inserito, e visualizza l'indirizzo precedente.
[ <i>Nuovo Dato = BYTE</i> ] < <i>Invio</i> >	Sostituisce il dato, nella locazione attuale, con il nuovo byte inserito, e visualizza l'indirizzo successivo.
< <i>Esc</i> >	Ritorna al prompt del Monitor/Debugger.

**NOTA BENE** A seguito della sostituzione di un byte, viene effettuata la verifica sull'operazione eseguita; nel caso di errore il Monitor/Debugger 188 suppone, che l'indirizzo in questione risieda in una zona dotata di circuiteria di protezione della scrittura (come la RAM-Disk sulle schede **GPC® 188F** e **GPC® 188D**). Quindi chiede una conferma prima di ripetere l'operazione, gestendo opportunamente la protezione in scrittura (necessaria per scrivere un byte in tale area di memoria); terminata questa operazione, non viene più effettuato nessun controllo.

**Esempio** Se le locazioni di memoria 01000H e 01001H (0100:0000 e 0100:0001), della RAM di lavoro, contengono il byte FFH e si desidera sostituire tali valori, con rispettivamente i dati 55H e AAH, sarà necessario inserire:

```
M188> SUBSTITUE AT 0100:0000 <Invio>
S> 0100:0000 = FF WITH 55 <Invio>
S> 0100:0001 = FF WITH AA <Invio>
S> 0100:0002 = xx WITH <Esc>
```

## ESECUZIONE DI UNA O PIU' ISTRUZIONI

*Sintassi:*            **T**    [ *Numero Step = WORD* ]            < *Invio* >

Esegue il numero di istruzioni di programma, indicato in "Numero Step" a partire dall'indirizzo esteso, determinato dai registri **CS:IP**.

Al termine dell'esecuzione di ogni step, vengono visualizzati tutti i vari registri ed il contenuto, deassemblato, della successiva istruzione del programma.

**Esempio** Se si vogliono eseguire tre istruzioni dell'applicativo, sarà necessario inserire:

```
M188> TRACE 0003 <Invio>
```



**TABELLA RIASSUNTIVA DEI COMANDI**

Di seguito viene riportata la tabella riassuntiva dei comandi del Monitor/Debugger 188.

COMANDO	TASTO	PARAMETRI
Assemblatore in Linea	A	[Inizio = IND. ESTESO] <Invio>
Esecuzione del programma in E000H	B	----
Copia di un blocco di memoria	C	[Inizio Blocco = IND. ESTESO] <Invio> [Fine Blocco = OFFSET] <Invio> [Destinazione = IND. ESTESO] <Invio>
Visualizzazione di un blocco di memoria	D	[Inizio Blocco = IND. ESTESO] <Invio> [Fine Blocco = OFFSET] <Invio>
Input di una WORD	E	[Indirizzo Periferica = WORD] <Invio>
Riempimento di un blocco di memoria	F	[Inizio Blocco = IND. ESTESO] <Invio> [Fine Blocco = OFFSET] <Invio> [Dato = BYTE] <Invio>
Esecuzione di un blocco di programma	G	[Inizio Blocco Prog. = IND. ESTESO] <Invio> [Fine Blocco Prog. = IND. ESTESO] <Invio>
Visualizzazione elenco comandi	H	----
Input di un BYTE	I	[Indirizzo Periferica = WORD] <Invio>
Esecuzione del programma in F000H	J	----
Deassembla un blocco di memoria	L	Inizio Blocco = IND. ESTESO] <Invio> [Fine Blocco = OFFSET] <Invio>
Output di un BYTE	O	[Indirizzo periferica = WORD] <Invio> [Dato = BYTE] <Invio>
Esecuzione di una Procedura	P	<Invio>
Riavvia il GDOS188	Q	----
Visualizzazione o modifica dei registri	R	[Nome Registro (opzionale)] <Invio>
Sostituzione di dati in memoria	S	[Inizio = IND. ESTESO] <Invio>
Esecuzione di una o più istruzioni	T	[Numero Step = WORD] <Invio>
Output di una WORD	U	[Indirizzo periferica = WORD] <Invio> [Dato = WORD] <Invio>
Visualizzazione in formato HEX Intel	W	[Inizio Blocco = IND. ESTESO] <Invio> [Fine Blocco = OFFSET] <Invio>
Ricezione di un file	X	<Spazio (selezione del tipo di file)> <Invio>

**FIGURA 30: TABELLA RIASSUNTIVA DEI COMANDI DEL MONITOR/DEBUGGER 188**

## DESCRIZIONE DEL PROGRAMMA TOOLS 188

Il pacchetto **GDOS 188**, comprende un programma denominato **TOOLS 188**; questo é un software che facilita la gestione di alcune risorse di bordo della scheda remota; infatti é possibile formattare la RAM-Disk, copiare dei file da P.C. a RAM-Disk o viceversa, rinominare o cancellare dei file, settare il Real-Time-Clock, ecc. Da notare che tutte le varie operazioni, che comportano una manipolazione di file sul P.C., saranno possibili solo nel caso in cui su quest'ultimo, vi sia in esecuzione l'emulazione terminale intelligente del programma **GET188.EXE**.

Di seguito viene illustrato come mandare in esecuzione il programma TOOLS 188, i vari comandi disponibili ed una breve descrizione degli stessi.

### ESECUZIONE DEL PROGRAMMA TOOLS 188

Il programma TOOLS 188, come indicato nel paragrafo "ORGANIZZAZIONE DELLA EPROM O FLASH EPROM", é memorizzato nella zona a bassa priorit  di AUTORUN (indirizzo **F0000H=F000:0000**); per eseguirlo é quindi necessario eseguire le seguenti operazioni:

- 1) Collegare la linea seriale A (console) della scheda ad una linea seriale del P.C., sfruttando il cavo di comunicazione descritto nel paragrafo "CAVO DI COMUNICAZIONE SERIALE".
- 2) Chiudere il jumper di selezione del modo operativo per attivare la modalit  Debug, come incato nel paragrafo "SELEZIONE DELLA MODALITA' RUN / DEBUG".
- 3) Lanciare il **GET188** sul P.C. di sviluppo collegato alla scheda.
- 4) Selezionare la giusta configurazione della linea seriale del P.C. (Com e Baud Rate), in accordo con l'attuale configurazione hardware della scheda, tramite l'apposita opzione "Options | Serial Port" del **GET188**.
- 5) Attivare l'emulazione terminale selezionando l'opzione "Options | Terminal" del **GET188**.
- 6) Resettare od accendere la scheda remota in modo da eseguire il Monitor/Debugger 188.
- 7) Fornire il comando di esecuzione del programma in F0000H con itasto "J".
- 8) A questo punto il programma TOOLS 188 parte, presentando le seguenti informazioni:

**TOOLS 188 - Ver x.x - GPC° ???? ff MHz**

**CHANGE DRIVE**  
COPY FILE  
DELETE FILE from M:  
RENAME FILE on M:  
FORMAT M:  
SET RTC  
EXIT

<Informazioni relative alla RAM-Disk M>

Dove con la indicazione generica composta da **x**, si intende il numero di versione, con **????** la scheda remota utilizzata e con **ff** la frequenza della stessa; mentre "Informazioni relative alla RAM-Disk M", é relativa ad una serie di indicazioni, in cui é riportato lo stato della RAM-Disk (se presente, se formattata, numero di KBytes disponibili, ecc.) ed eventualmente la directory dei file contenuti.

## COMANDI DEL PROGRAMMA TOOLS 188

Il programma TOOLS 188, una volta in esecuzione, mette a disposizione una serie di comandi, selezionabili in modo molto rapido ed intuitivo.

Infatti, mediante i tasti <Freccia Up> e <Freccia Down>, é possibile muoversi all'interno del menù principale, evidenziando il comando che si desidera utilizzare, quindi confermarlo con <Invio>. Ogni operazione selezionata, può essere abortita, durante la fase di inserimento dei parametri, mediante il tasto <Esc>.

Di seguito viene riportata una descrizione di tutti vari comando che si possono eseguire, oltre ai vari parametri ad essi necessari.

### **SELEZIONE DRIVE E VISUALIZZAZIONE DIRECTORY**

Viene eseguita selezionando il comando **CHANGE DRIVE** dal menù principale e consente di selezionare il drive, fra quelli del P.C. o la RAM-Disk, quindi di visualizzarne il relativo contenuto. Per eseguire tale operazione é necessario specificare la lettera associata al drive desiderato (**Enter DRIVE**); sarà quindi visualizzato l'elenco dei file presenti nella directory attuale del disco richiesto. Questo é organizzato in pagine di **45 file**, le quali si possono scorrere mediante i tasti <Pg Up> e <Pg Down>.

#### **N.B.**

Questo comando è utilizzabile correttamente solo nel caso in cui sul P.C., vi sia in esecuzione l'emulazione terminale intelligente del programma **GET188.EXE**.

### **COPIA DI FILE**

Viene eseguita selezionando il comando **COPY FILE** dal menù principale e consente di copiare uno più file da due drive, fra quelli del P.C. o la RAM-Disk.

Per eseguire la copia di file, é necessario, innanzi tutto, inserire le lettere relative al disco sorgente (**SOURCE DRIVE**) e destinazione (**DEST DRIVE**); a questo punto, verrà visualizzata la directory corrente del disco sorgente e mediante i tasti <Freccia Up>, <Freccia Down>, <Pg Up>, <Pg Down> ed <Invio>, si potranno selezionare ed evidenziare tutti i file che si dederà copiare, quindi eseguire l'operazione con il tasto <G>. Se alcuni di questi sono già presenti sul disco destinazione, sarà visualizzata una richiesta di conferma, prima di riscrivere il file in questione.

#### **N.B.**

Questo comando è utilizzabile correttamente solo nel caso in cui sul P.C., vi sia in esecuzione l'emulazione terminale intelligente del programma **GET188.EXE**, inoltre la copiare di file in RAM-Disk, é possibile solo se questa é presente e formattata.

## CANCELLAZIONE DI FILE DA RAM-DISK

Viene eseguita selezionando il comando **DELETE FILE from M:** dal menù principale e consente di cancellare uno più file dalla RAM-Disk della scheda remota.

Eseguendo questo comando, verrà visualizzata la directory corrente del drive M (RAM-Disk) e mediante i tasti <Freccia Up>, <Freccia Down>, <Pg Up>, <Pg Down> ed <Invio>, si potranno selezionare ed evidenziare i vari file che si desidera cancellare, quindi eseguire l'operazione con il tasto <G>, a cui segue una richiesta di conferma.

## VARIAZIONE DEL NOME DI UN FILE SU RAM-DISK

Viene eseguita selezionando il comando **RENAME FILE on M:** dal menù principale e consente di rinominare un file presente nella RAM-Disk della scheda remota.

Eseguendo questo comando, verrà visualizzata la directory corrente del drive M (RAM-Disk) e mediante i tasti <Freccia Up>, <Freccia Down>, <Pg Up>, <Pg Down> ed <Invio>, si potrà selezionare ed evidenziare il file che si desidera rinominare, quindi eseguire l'operazione con il tasto <G>, a cui segue la richiesta del nuovo nome da assegnare al file selezionato.

### N.B.

Questo comando sarà eseguito solo se viene selezionato un solo file.

## FORMATTAZIONE DELLA RAM-DISK

Viene eseguita selezionando il comando **FORMAT M:** dal menù principale e consente di formattare la RAM-Disk della scheda remota, in modo da poterla utilizzare, in tutte le varie operazioni, come un normale disco.

Eseguendo questo comando, sarà visualizzato un messaggio di conferma dell'operazione, se la RAM-Disk è già esistente, quindi verrà richiesta l'ampiezza, in KByte della stessa (**Enter KBYTE RAM-Disk size (1..256):** ).

Tale dato non deve superare la quantità di SRAM che il **GDOS 188** ha destinato a RAM-Disk, in caso contrario sarà visualizzato un errore.

Al termina della formattazione viene visualizzato un messaggio, in cui è riportato l'esito dell'operazione e la segnalazione di eventuali errori.

## SETTAGGIO DEL REAL TIME CLOCK

Viene eseguita selezionando il comando **SET RTC** dal menù principale e consente di inizializzare, in termini di data e ora, il Real Time Clock, presente a bordo della scheda remota.

Eseguendo questa operazione, verrà visualizzata, innanzi tutto, la data attuale, secondo il formato:

**ww dd/mm/yy**

Dove con **ww**, **dd**, **mm** e **yy** si intendono rispettivamente il giorno della settimana, il giorno, il mese e l'anno. A questo punto sarà possibile confermare tale data, e passare al settaggio dell'ora, oppure inserire i nuovi valori.

In questo secondo caso, verrà innanzi tutto richiesto l'inserimento del giorno (**dd**), del mese (**mm**) e dell'anno (**yy**) nel formato:

**dd/mm/yy**

seguito dalla richiesta di inserimento del giorno della settimana da effettuare con i seguenti tasti:

**0 -> Domenica (Su)**

**1 -> Lunedì (Mo)**

**2 -> Martedì (Tu)**

**3 -> Mercoledì (We)**

**4 -> Giovedì (Th)**

**5 -> Venerdì (Fr)**

**6 -> Sabato (Sa)**

Successivamente verrà visualizzata l'ora attuale, secondo il formato:

**hh:mm:ss**

Dove con **hh**, **mm** e **ss** si intendono rispettivamente l'Ora, i Minuti ed i Secondi. A questo punto sarà possibile confermare quanto visualizzato, e tornare al menù principale, oppure inserire i nuovi valori, nel formato sopra descritto.

## USCITA DAL PROGRAMMA TOOLS 188

Viene eseguita selezionando il comando **EXIT**, oppure premendo il tasto **<Esc>**, dal menù principale e permette di terminare l'esecuzione del programma TOOLS 188, quindi ritornare al Monitor/Debugger 188.



## FUNZIONI DEL SISTEMA OPERATIVO GDOS 188

Il sistema operativo **GDOS 188** è provvisto di una serie di funzioni di base, tramite le quali l'utente si interfaccia direttamente all'hardware del sistema, senza doverne conoscere le caratteristiche. Ogni procedura, è direttamente utilizzabile durante la programmazione ed ha propri parametri d'ingresso e d'uscita, ma tutte sono accomunate dal modo in cui vengono richiamate:

- 1) Inizializzazione parametri
- 2) Selezione funzione tramite il registro **AH** della CPU
- 3) Chiamata al vettore di Interrupt **3BH** o **21H**.
- 4) La procedura ritorna automaticamente al punto di chiamata.

L'utente per lo sviluppo del proprio applicativo può utilizzare in qualsiasi momento tutte le varie funzioni a disposizione, devono essere però ripetute le seguenti regole:

- 1) Le procedure in cui vi sono delle chiamate a funzioni del **GDOS 188** non devono essere ricorsive.
- 2) Non devono essere modificati i vettori di Interrupt **3BH** e **21H**.
- 3) Non deve essere alterato il valore del registro **PBA** della CPU.

### N.B.

Molte funzioni del sistema operativo **GDOS 188**, sono già utilizzate anche dalle varie istruzioni del compilatore PASCAL 188 quali ad esempio Write(...), WriteLn(...), Write(AUX,...), Write(LST,...), Assign ....., Write(<file>,...), ClrScr, ClrEol, GotoXY(...), ecc.

Quindi sviluppando un applicativo con i comandi ad alto livello, tipici di questo ambiente, non è necessario effettuare delle chiamate a tali procedure; l'uso di queste funzioni è invece di particolare interesse, a tutti gli utenti che intendono sviluppare software di gestione in assembly, in quanto non devono implementare i vari driver d'interfacciamento con l'hardware della scheda remota.

Le procedure relative alla gestione di RAM-Disk, EEPROM ed RTC, sono invece utili per qualsiasi ambiente di programmazione, in quanto permettono di ridurre la quantità di codice sorgente e di utilizzare il dispositivo in questione, con una gestione ad alto livello.

Le funzioni del sistema operativo **GDOS 188**, sono divise in due gruppi, disponibili attraverso delle chiamate al vettore di Interrupt **3BH** nel primo caso, oppure al vettore **21H** nel secondo caso.

Nelle pagine seguenti, ne è riportata una descrizione completa.

## FUNZIONI DEL VETTORE DI INTERRUPT 3BH

Le funzioni del sistema operativo **GDOS 188**, riportate di seguito, sono quelle disponibili attraverso una chiamata al **vettore di Interrupt 3BH**; il numero della procedura richiesta deve essere memorizzato nel registro **AH**, mentre altri registri, specificati di volta in volta, sono utilizzati per il passaggio dei vari parametri.

Nel caso in cui l'utente sviluppi il proprio programma in linguaggio assembly, per chiamare le varie funzioni, deve inserire le seguenti istruzioni:

<b>MOV</b>	<b>AH, &lt;Numero Funzione&gt;</b>	
<b>MOV</b>	<b>.....</b>	; Settaggio altri eventuali registri
<b>INT</b>	<b>3BH</b>	; Chiamata al vettore INT 3BH



## FUNZIONE 2: INPUT DA CONSOLE

*Input:*     **AH = 2**  
*Output:*   **AL = <Dato ricevuto>**

Attende la ricezione di un carattere dalla linea di console (linea seriale A) della scheda remota, e lo restituisce nel registro **AL**.

## FUNZIONE 3: OUTPUT SU CONSOLE

*Input:*     **AH = 3**  
              **AL = <Dato da trasmettere>**  
*Output:*    **Nessuno**

Trasmette il carattere memorizzato nel registro **AL**, sulla linea di console (linea seriale A) della scheda remota.

## FUNZIONE 4: STATO DELLA LINEA AUSILIARIA

*Input:*     **AH = 4**  
*Output:*    **ZERO flag = Stato del ricevitore     ( Z = 0 -> Ricevuto un carattere )**  
              **CARRY flag = Stato del trasmettitore   ( C = 1 -> Pronto a trasmettere )**

Controlla il registro di stato della linea ausiliaria (linea seriale B) e riporta le informazioni su ricevitore e trasmettitore, rispettivamente nei flag **ZERO** e **CARRY**.

## FUNZIONE 15: OUTPUT SU CONSOLE DI "BELL"

*Input:*     **AH = 15 (0FH)**  
*Output:*    **Nessuno**

Trasmette il codice ASCII di "BELL" (**07**), sulla linea di console (linea seriale A) della scheda remota.

## FUNZIONE 16: OUTPUT SU CONSOLE DI "CR+LF"

*Input:*     **AH = 16 (10H)**  
*Output:*    **Nessuno**

Trasmette i codici ASCII di "CR"+"LF" (**13** e **10**), sulla linea di console (linea seriale A) della scheda remota.

## FUNZIONE 20: POSIZIONAMENTO DEL CURSORE

*Input:*     **AH = 20 (14H)**  
              **DH = <Riga>**  
              **DL = <Colonna>**  
*Output:*    **Nessuno**

Trasmette, sulla linea di console della scheda remota, i codici di posizionamento del cursore alla posizione di riga e colonna, indicata nei registri **DH** e **DL** (**0,0** corrisponde alla posizione di **Home**). La sequenza dei codici trasmessa, é quella relativa al programma **GET188**, ed é indicata nel paragrafo "EMULAZIONE TERMINALE".

### **N.B.**

Questa funzione sar  eseguita correttamente, solo se la scheda remota   collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 21: CLEAR SCREEN

*Input:*     **AH = 21 (15H)**  
*Output:*    **Nessuno**

Trasmette, sulla linea di console della scheda remota, i codici di **clear screen** (cancellazione dello schermo). La sequenza   quella relativa al programma **GET188**, ed   indicata nel paragrafo "EMULAZIONE TERMINALE".

### **N.B.**

Questa funzione sar  eseguita correttamente, solo se la scheda remota   collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 22: CLEAR END OF LINE

*Input:*     **AH = 22 (16H)**  
*Output:*    **Nessuno**

Trasmette, sulla linea di console della scheda remota, i codici di **clear end of line** (cancellazione dalla posizione del cursore fino a fine linea). La sequenza   quella relativa al programma **GET188**, ed   indicata nel paragrafo "EMULAZIONE TERMINALE".

### **N.B.**

Questa funzione sar  eseguita correttamente, solo se la scheda remota   collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

### FUNZIONE 23: DELETE LINE

*Input:*     **AH = 23 (17H)**

*Output:*    *Nessuno*

Trasmette, sulla linea di console della scheda remota, i codici di **delete line** (cancellazione della linea puntata dal cursore). La sequenza é quella relativa al programma **GET188**, ed é indicata nel paragrafo "EMULAZIONE TERMINALE".

#### **N.B.**

Questa funzione sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

### FUNZIONE 24: INSERT LINE

*Input:*     **AH = 24 (18H)**

*Output:*    *Nessuno*

Trasmette, sulla linea di console della scheda remota, i codici di **insert line** (inserimento di una linea nella posizione del cursore). La sequenza é quella relativa al programma **GET188**, ed é indicata nel paragrafo "EMULAZIONE TERMINALE".

#### **N.B.**

Questa funzione sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

### FUNZIONE 25: ALTA LUMINOSITA'

*Input:*     **AH = 25 (19H)**

*Output:*    *Nessuno*

Trasmette, sulla linea di console della scheda remota, i codici dell'attributo di **highvideo** (alta luminosità). La sequenza é quella relativa al programma **GET188**, ed é indicata nel paragrafo "EMULAZIONE TERMINALE".

#### **N.B.**

Questa funzione sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 26: LUMINOSITA' NORMALE

*Input:*     **AH = 26 (IAH)**

*Output:*    **Nessuno**

Trasmette, sulla linea di console della scheda remota i codici dell'attributo di **lowvideo** (luminosità normale). La sequenza é quella relativa al programma **GET188**, ed é indicata nel paragrafo "EMULAZIONE TERMINALE".

### **N.B.**

Questa funzione sar  eseguita correttamente, solo se la scheda remota   collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 31: ABILITA LA FORMATTAZIONE DELLA RAM-DISK

*Input:*     **AH = 31 (IFH)**

*Output:*    **Nessuno**

Abilita l'operazione di formattazione della RAM-Disk montata sulla scheda remota (drive **M:**). Solo dopo questa operazione,   possibile chiamare la funzione di formattazione vera e propria (funzione **32**).

## FUNZIONE 32: FORMATTAZIONE DELLA RAM-DISK

*Input:*     **AH = 32 (20H)**

**AL = <Ampiezza in KByte>**

*Output:*    **AL = <Esito della formattazione>**

Effettua l'operazione di formattazione della RAM-Disk, montata sulla scheda remota (drive **M:**), l'ampiezza in KByte, indicata nel registro **AL**, secondo la seguente corrispondenza:

**AL = 1÷255**     ->   1÷255 KBytes

**AL = 0**         ->   256 KBytes

Sempre nel registro **AL** viene restituito l'esito di tale inizializzazione, in particolare:

**AL = 0**     ->   Formattazione conclusa correttamente.

**AL = 1**     ->   Operazione di formattazione disabilitata (chiamare la funzione **31**).

**AL = 2**     ->   Errore durante l'operazione di formattazione.

**AL = 3**     ->   Errore durante l'operazione di verifica della formattazione.

**AL = 4**     ->   Spazio di RAM insufficiente per formattare la quantit  di KBytes richiesta.

### **N.B.**

Prima di effettuare la formattazione della RAM-Disk, deve sempre essere chiamata la funzione **31**, descritta in precedenza.

### FUNZIONE 33: CONTROLLO DELLA RAM-DISK

*Input:*     **AH = 33 (21H)**  
*Output:*    **AL = <Esito delle verifiche>**

Effettua una serie di verifiche sulla RAM-Disk, montata sulla scheda remota (drive **M:**), e restituisce l'esito nel registro **AL**, in particolare:

**AL = 0**     -> RAM-Disk valida e corretta.  
**Bit 0 AL = 1**   -> RAM-Disk non presente o non formattata.  
**Bit 1 AL = 1**   -> Directory della RAM-Disk incongruente.  
**Bit 2 AL = 1**   -> Operazione di write in RAM-Disk non conclusa.

### FUNZIONE 34: LETTURA DI UN BYTE DALL'EEPROM

*Input:*     **AH = 34 (22H)**  
              **BX = <Indirizzo da leggere>**  
*Output:*    **AH = <Esito della lettura>**  
              **AL = <Byte letto>**

Effettua la lettura di un byte dall'EEPROM di bordo della scheda remota, all'indirizzo indicato nel registro **BX**.

Nel registri **AL** ed **AH** vengono restituiti rispettivamente il byte letto e l'esito dell'operazione, secondo la seguente corrispondenza:

**AH = 0**     -> Lettura conclusa correttamente.  
**AH = 255**  -> Errore durante la lettura.

Da notare che in caso di errore di lettura, il valore restituito in **AL** non é significativo.

#### **N.B.**

Questa procedura non effettua nessuna verifica sull'indirizzo da leggere in relazione alla quantità di EEPROM installata, quindi l'utente deve preoccuparsi di non fornire dei valori al di fuori dell'ultima locazione disponibile.

### FUNZIONE 35: SCRITTURA DI UN BYTE NELL'EEPROM

*Input:*     **AH** = 35 (23H)  
              **AL** = <Byte da scrivere>  
              **BX** = <Indirizzo da scrivere>  
*Output:*    **AH** = <Esito della scrittura>

Effettua la scrittura del byte contenuto in **AL**, nell'EEPROM di bordo della scheda remota, all'indirizzo indicato nel registro **BX**.

Nel registro **AH** viene restituito l'esito dell'operazione, secondo la seguente corrispondenza:

**AH** = 0     ->   Scrittura conclusa correttamente.  
**AH** = 255  ->   Errore durante la scrittura.

In caso di operazione conclusa correttamente, questa procedura genera anche il ritardo di **10 msec**, necessario ad assicurare una corretta memorizzazione del byte nella EEPROM.

#### **N.B.**

Questa procedura non effettua nessuna verifica sull'indirizzo da scrivere in relazione alla quantità di EEPROM installata, quindi l'utente deve preoccuparsi di non fornire dei valori al di fuori dell'ultima locazione disponibile.

### FUNZIONE 36: SCRITTURA NELLA SRAM PROTETTA IN SCRITTURA

*Input:*     **AH** = 36 (24H)  
              **AL** = <Byte da scrivere>  
              **ES:BX** = <Indirizzo da scrivere>  
*Output:*    Nessuno

Effettua la scrittura del byte indicato in **AL**, nella zona di SRAM, dotata di protezione in scrittura, all'indirizzo indicato nella coppia di registri **ES:BX**.

Questa funzione é di particolare interesse quando si deve gestire direttamente l'area di SRAM, normalmente destinata a RAM-Disk, dotata di protezione in scrittura (schede **GPC® 188F** e **GPC® 188D**), in quanto é la procedura stessa che gestisce opportunamente tale circuiteria prima di effettuare la scrittura del byte.

#### **N.B.**

Per motivi di compatibilità, questa funzione é sempre disponibile, in tutte le versioni del **GDOS 188**, anche in quelle destinate a schede prive della circuiteria di protezione della scrittura.

In questo caso però, questa procedura esegue una semplice scrittura in memoria, quindi non é conveniente il suo utilizzo.



## TABELLA RIASSUNTIVA DELLE FUNZIONI DELL'INTERRUPT 3BH

Di seguito viene riportata la tabella riassuntiva delle funzioni relative al vettore di interrupt **3BH**.

FUNZIONE	INPUT	OUTPUT
<b>Acquisizione di un tasto</b>	AH = 0	AL = <Codice tasto> oppure: AL = 0 AH = <Codice tasto speciale>
<b>Stato della console</b>	AH = 1	Z = 0 (Ricevuto un carattere) C = 1 (Pronto a trasmettere)
<b>Input da console</b>	AH = 2	AL = <Dato Ricevuto>
<b>Output su console</b>	AH = 3 AL = <Dato da trasmettere>	----
<b>Stato della linea ausiliaria</b>	AH = 4	Z = 0 (Ricevuto un carattere) C = 1 (Pronto a trasmettere)
<b>Output su console di "BELL"</b>	AH = 15 (0FH)	----
<b>Output su console di "CR+LF"</b>	AH = 16 (10H)	----
<b>Posizionamento cursore</b>	AH = 20 (14H) DH = <Riga> DL = <Colonna>	----
<b>Clear Screen</b>	AH = 21 (15H)	----
<b>Clear End of Line</b>	AH = 22 (16H)	----
<b>Delete line</b>	AH = 23 (17H)	----
<b>Insert Line</b>	AH = 24 (18H)	----
<b>Alta Luminosità</b>	AH = 25 (19H)	----
<b>Luminosità Normale</b>	AH = 26 (1AH)	----
<b>Abilita la formattazione della RAM-Disk</b>	AH = 31 (1FH)	----
<b>Formattazione RAM-Disk</b>	AH = 32 (20H) AL = <Ampiezza (KByte)>	AL = <Esito formattazione>
<b>Controllo RAM-Disk</b>	AH = 33 (21H)	AL = <Esito delle verifiche>
<b>Lettura da EEPROM</b>	AH = 34 (22H) BX = <Indirizzo>	AL = <Byte letto> AH = <Esito della lettura>
<b>Scrittura in EEPROM</b>	AH = 35 (23H) AL = <Byte da scrivere> BX = <Indirizzo>	AH = <Esito della scrittura>
<b>Scrittura nella RAM protetta in scrittura</b>	AH = 36 (24H) AL = <Byte da scrivere> ES:BX = <Indirizzo>	----

FIGURA 31: TABELLA DELLE FUNZIONI DELL'INTERRUPT 3BH

## FUNZIONI DEL VETTORE DI INTERRUPT 21H

Le funzioni del sistema operativo **GDOS 188**, riportate di seguito, sono quelle disponibili attraverso una chiamata al **vettore di Interrupt 21H**; il numero della procedura richiesta deve essere memorizzato nel registro **AH**, mentre altri registri, specificati di volta in volta, sono utilizzati per il passaggio dei vari parametri.

Nel caso in cui l'utente sviluppi il proprio programma in linguaggio assembly, per chiamare le varie funzioni, deve inserire le seguenti istruzioni:

<b>MOV</b>	<b>AH, &lt;Numero Funzione&gt;</b>	
<b>MOV</b>	<b>.....</b>	; Settaggio altri eventuali registri
<b>INT</b>	<b>21H</b>	; Chiamata al vettore INT 21H

Se invece si sviluppa un programma con il compilatore PASCAL 188, dovranno essere inserite, ad esempio, le seguenti istruzioni:

```

type RegPack = record
    AX,BX,CX,DX,BP,DI,SI,DS,ES,Flags : integer;
end;

var Reg : RegPack;

.....

Reg.AX := (<Numero funzione> shl 8) + <AL>; { Settaggio di AH ed AL }
Reg.?? := ..... { Settaggio altri eventuali registri }
msdos( Reg); { Chiamata al vettore INT 21H }
    
```

Esempi espliciti di chiamate alla procedure dell'interrupt 3BH sono riportati nei vari programmi di esempio, contenuti all'interno del disco di distribuzione.

Di seguito é riportato l'elenco completo relativo alle procedure dell'interrupt 21H.

### **FUNZIONE 0: TERMINA L'ESECUZIONE DEL PROGRAMMA**

*Input:*     **AH = 0**  
*Output:*    **Nessuno**

Termina l'esecuzione del programma corrente, restituendo il controllo al **Monitor/Debugger 188**.

### **FUNZIONE 3: INPUT DALLA LINEA AUSILIARIA**

*Input:*     **AH = 3**  
*Output:*    **AL = <Dato ricevuto>**

Attende la ricezione di un carattere dalla linea di ausiliaria (linea seriale B) della scheda remota, e lo restituisce nel registro **AL**.

#### FUNZIONE 4: OUTPUT SU LINEA AUSILIARIA

*Input:*     **AH = 4**  
              **DL = <Dato da trasmettere>**  
*Output:*    **Nessuno**

Trasmette il carattere memorizzato nel registro **DL**, sulla linea ausiliaria (linea seriale A) della scheda remota.

#### FUNZIONE 5: OUTPUT SU STAMPANTE LOCALE

*Input:*     **AH = 5**  
              **DL = <Dato da stampare>**  
*Output:*    **Nessuno**

Stampa il carattere memorizzato nel registro **DL**, sulla stampante parallela, di tipo CENTRONICS, collegata al connettore di I/O della scheda remota.

#### FUNZIONE 13: SVUOTA I BUFFER DEI FILE

*Input:*     **AH = 13 (0DH)**  
*Output:*    **Nessuno**

Permette di svuotare tutti i buffer interni dei file nel disco corrente (RAM-Disk, oppure drive del P.C.). Questa funzione, però non aggiorna le voci della directory, né chiude i file aperti.

#### **N.B.**

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 14: SELEZIONE DEL DRIVE CORRENTE

*Input:*     **AH = 14 (0EH)**  
              **DL = <Numero del drive>**  
*Output:*    **AL = <Numero unità installate>**

Seleziona il nuovo drive corrente; riporta inoltre il numero delle unità installate. L'unità disco viene specificata nel registro **DL**, in particolare:

<b>DL = 0</b>	->	Drive <b>A</b> del P.C.
<b>DL = 1</b>	->	Drive <b>B</b> del P.C.
<b>DL = 2</b>	->	Drive <b>C</b> del P.C. (Hard Disk)
<b>DL = 3</b>	->	Drive <b>D</b> del P.C. (Hard-Disk, CD-ROM o unità removibile)
.....		
<b>DL = 12 (0CH)</b>	->	Drive <b>M</b> (RAM-Disk)

Nel registro **AL**, viene restituito il numero delle unità disco intallate; nel caso in cui si selezioni come drive corrente il disco **M**, la funzione restituisce nel registro **AL** il valore **1**.

### **N.B.**

Questa funzione può comportare l'elaborazione di dati sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 25: ACQUISIZIONE DEL DRIVE CORRENTE

*Input:*     **AH = 25 (19H)**  
*Output:*    **AL = <Drive corrente>**

Riporta in **AL**, il lettore corrente, usando il codice numerico standard, illustrato in precedenza: drive A=0, drive B=1, drive C=2, drive D=3, ....., drive M=12.

### **N.B.**

Questa funzione può comportare l'elaborazione di dati sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 26: SETTA L'INDIRIZZO DELLA DTA

*Input:*     **AH = 26 (1AH)**  
              **DS:DX = <Indirizzo DTA>**  
*Output:*    **Nessuno**

Permette di stabilire l'indirizzo dell'area di trasferimento del disco (**Disk Transfer Area: DTA**), che sarà usata per le operazione di I/O sui file. L'indirizzo di quest'area di memoria, viene specificata dalla coppia di registri **DS:DX**. E' necessario specificare un indirizzo della **DTA**, prima di usare qualsiasi funzione dell'interrupt **21H** che utilizza tale area.

**FUNZIONE 42: ACQUISIZIONE DELLA DATA**

*Input:*     **AH = 42 (2AH)**  
*Output:*   **AL = <Giorno della settimana>**  
              **CH = 0**  
              **CL = <Anno>**  
              **DH = <Mese>**  
              **DL = <Giorno>**

Viene acquisita la data corrente, dal Real Time Clock della scheda remota, e restituita nei registri **AL**, **CX** e **DX**. **DH** contiene il numero del mese (da 1 a 12); **DL** contiene il giorno del mese (da 1 a 28, 29, 30 o 31 secondo i casi); **CL** contiene l'anno (da 00 a 99); **CH** restituisce sempre il valore **0**; infine **AL** contiene il giorno della settimana (0 = Domenica, 1 = Lunedì, ..., 6 = Sabato).

**FUNZIONE 43: SETTAGGIO DELLA DATA**

*Input:*     **AH = 43 (2BH)**  
              **AL = <Giorno della settimana>**  
              **CL = <Anno>**  
              **DH = <Mese>**  
              **DL = <Giorno>**  
*Output:*   **AL = 0**

Viene settato il Real Time Clock della scheda remota, con la data indicata nei registri **AL**, **CL** e **DX**. **DH** contiene il numero del mese (da 1 a 12); **DL** contiene il giorno del mese (da 1 a 28, 29, 30 o 31 secondo i casi); **CL** contiene l'anno (da 00 a 99); infine **AL** contiene il giorno della settimana (0 = Domenica, 1 = Lunedì, ..., 6 = Sabato).

Questa procedura inoltre restituisce il valore **0** in **AL**.

**FUNZIONE 44: ACQUISIZIONE DELL'ORA**

*Input:*     **AH = 44 (2CH)**  
*Output:*   **CH = <Ore>**  
              **CL = <Minuti>**  
              **DH = <Secondi>**

Viene acquisita l'ora corrente, dal Real Time Clock della scheda remota, e restituita nei registri **CX** e **DH**. **CH** contiene l'ora (da 0 a 23, sulla base di un orologio a 24 ore); **CL** contiene i minuti (da 0 a 59); e **DH** contiene i secondi (da 0 a 59).

## FUNZIONE 45: SETTAGGIO DELL'ORA

*Input:*     **AH = 45 (2DH)**  
           **CH = <Ore>**  
           **CL = <Minuti>**  
           **DH = <Secondi>**  
*Output:*    **AL = 0**

Viene settato il Real Time Clock della scheda remota, con l'ora indicata nei registri **CX** e **DH**. **CH** contiene l'ora (da 0 a 23, sulla base di un orologio a 24 ore); **CL** contiene i minuti (da 0 a 59); e **DH** contiene i secondi (da 0 a 59). Questa procedura inoltre restituisce il valore **0** in **AL**.

## FUNZIONE 48: ACQUISIZIONE DEL NUMERO DI VERSIONE

*Input:*     **AH = 48 (30H)**  
*Output:*    **AX = <Numero versione>**  
           **BX: CX = <Numero di serie>**

Viene restituito, nel registro **AX**, il numero di versione del sistema operativo **GDOS 188**. Questo sarà sempre **0001** (corrispondente alla release **1.0**), per compatibilità con il software esistente. I registri **BX** e **CX** contengono il numero di serie, che, per lo stesso motivo, sarà sempre **0**.

## FUNZIONE 54: ACQUISIZIONE DELLO SPAZIO LIBERO SU DISCO

*Input:*     **AH = 54 (36H)**  
           **DL = <Numero drive>**  
*Output:*    **AX = <Settori per Cluster>**            **AX = FFFFH (Numero drive errato)**  
           **BX = <Cluster non utilizzati>**  
           **CX = <Byte per Settore>**  
           **DX = <Totale Cluster su disco>**

Vengono restituite una serie di informazioni relative all'unità disco indicata nel registro **DL**:

<b>DL = 0</b>	->	Drive corrente
<b>DL = 1</b>	->	Drive <b>A</b> del P.C.
<b>DL = 2</b>	->	Drive <b>B</b> del P.C.
<b>DL = 3</b>	->	Drive <b>C</b> del P.C. (Hard Disk)
<b>DL = 4</b>	->	Drive <b>D</b> del P.C. (Hard-Disk, CD-ROM o unità removibile)
.....		
<b>DL = 13 (0DH)</b>	->	Drive <b>M</b> (RAM-Disk)

Se si specifica il numero relativo ad un lettore non valido, questa funzione restituisce riporta **FFFFH** nel registro **AX**. Altrimenti **AX** contiene il numero di settori per unità di assegnazione o cluster, **CX** contiene il numero di byte per settore, **BX** contiene il numero di cluster disponibili e **DX** contiene il numero totale di cluster.

Con questi numeri si possono fare molti calcoli interessanti, nel modo seguente:

$$\begin{aligned}
 CX * AX &= \text{Byte per cluster} \\
 CX * AX * BX &= \text{Numero totale di byte liberi} \\
 CX * AX * DX &= \text{Spazio totale di immagazzinaggio in byte} \\
 (BX * 100) / DX &= \text{Percentuale di spazio libero}
 \end{aligned}$$

Se **S** é la grandezza di un file in byte, si può calcolare il numero di cluster occupati nel seguente modo:

$$(S + CX * AX - 1) / (CX * AX)$$

### N.B.

Questa funzione può comportare l'elaborazione di dati sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

### FUNZIONE 60: MAKE DI UN FILE

*Input:*     **AH = 60 (3CH)**  
               **CX = <Attributo>**  
               **DS:DX = <Ind. nome file in ASCIIZ>**  
*Output:*    **CARRY flag = 1**        **(Errore)**    **CARRY flag = 0**    **(Nessun Errore)**  
               **AX = <Codice errore>**            **AX = <Handle>**

Questa funzione apre un file nuovo usando un nome specificato. Se questo esiste viene troncato a lunghezza zero; viceversa viene creato un nuovo file con il nome e percorso specificato; questo deve essere fornito in formato **ASCIIZ** (stringa ASCII terminata dal codice **0**). La coppia di registri **DS:DX** contiene l'indirizzo della stringa in questione, mentre **CX** contiene i vari attributi del file (non gestiti in RAM-Disk), secondo seguente codifica:

**Bit 0 CX = 1**   ->   Setta l'attributo di sola lettura  
**Bit 1 CX = 1**   ->   Setta l'attributo di file Nascosto  
**Bit 2 CX = 1**   ->   Setta l'attributo di file Sistema  
**Bit 5 CX = 1**   ->   Setta l'attributo di file di Archivio

I restanti bit non sono gestiti e devono necessariamente essere posti a **0**.

Quando la funzione viene eseguita con successo, restituisce il **CARRY flag** azzerato e memorizza il codice di **Handle** relativo al file in questione, nel registro **AX**. Viceversa viene settato il **CARRY flag** e riportato in **AX** uno dei seguenti codici di errore:

**AX = 03**   ->   Percorso non trovato  
**AX = 04**   ->   Nessun codice di Handle disponibile  
**AX = 05**   ->   Accesso inibito (esaurito lo spazio disponibile o file di sola lettura)

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 61: OPEN DI UN FILE

*Input:*     **AH** = 61 (3DH)  
               **AL** = <Codice accesso>  
               **DS:DX** = <Ind. nome file in ASCIIZ>  
*Output:*    **CARRY flag** = 1        (Errore)   **CARRY flag** = 0    (Nessun Errore)  
               **AX** = <Codice errore>           **AX** = <Handle>

Questa funzione apre un file già esistente con nome e percorso indicati; questa stringa deve essere fornita in formato **ASCIIZ** (stringa ASCII terminata dal codice **0**). La coppia di registri **DS:DX** contiene l'indirizzo della stringa in questione, mentre il registro **AL** contiene un codice che indica come deve essere usato tale file, in particolare:

**AL** = 0     ->    Accesso al file solo in Lettura  
**AL** = 1     ->    Accesso al file solo in Scrittura  
**AL** = 2     ->    Accesso al file sia in Lettura che in Scrittura

I restanti valori del registro **AL** non sono gestiti e generano un'errore in fase di apertura del file. Quando la funzione viene eseguita con successo, restituisce il **CARRY flag** azzerato e memorizza il codice di **Handle** relativo al file in questione, nel registro **AX**. Viceversa viene settato il **CARRY flag** e riportato in **AX** uno dei seguenti codici di errore:

**AX** = 02     ->    File non trovato  
**AX** = 03     ->    Percorso non trovato  
**AX** = 04     ->    Nessun codice di Handle disponibile  
**AX** = 05     ->    Accesso inibito  
**AX** = 12 (0CH) ->    Codice di accesso non valido

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 62: CLOSE DI UN FILE

*Input:*     **AH** = 62 (3EH)  
               **BX** = <Handle>  
*Output:*    **CARRY flag** = 1        (Errore)   **CARRY flag** = 0    (Nessun Errore)  
               **AX** = <Codice errore>

Questa funzione chiude il file relativo al numero di handle fornito nel registro **BX**. Vengono svuotati tutti i buffer del file ad aggiornata, se necessario la directory; quindi, se eseguita con successo, restituisce il **CARRY flag** azzerato; viceversa viene settato tale flag e riportato in **AX** il codice di errore, che può essere solo **AX=6** (numero di handle non valido).

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.



## FUNZIONE 63: LEGGE DA UN FILE

*Input:*     **AH = 63 (3FH)**  
               **BX = <Handle>**  
               **CX = <Numero di byte da leggere>**  
               **DS:DX = <Ind. Buffer>**  
*Output:*    **CARRY flag = 1 (Errore)**    **CARRY flag = 0 (Nessun Errore)**  
               **AX = <Codice errore>**        **AX = <Numero di byte letti>**  
   **DS:DX = <Ind. Buffer con byte letti>**

Questa funzione legge il file associato al numero di Handle riportato in **BX**; il registro **CX** specifica il numero di byte che si devono leggere, mentre **DS:DX** puntano al buffer in cui saranno collocati i dati che vengono letti. Se l'operazione di lettura ha successo, viene azzerato il **CARRY flag** e restituito, in **AX**, il numero di byte letti (se **AX=0** la funzione ha tentato di leggere oltre la fine del file); viceversa viene settato il flag e riportato, sempre in **AX**, il codice di errore, in particolare:

**AX = 05**       ->    Accesso inibito (file aperto solo in scrittura)  
**AX = 06**       ->    Numero di handle non valido

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 64: SCRIVE SU UN FILE

*Input:*     **AH = 64 (40H)**  
               **BX = <Handle>**  
               **CX = <Numero di byte da scrivere>**  
               **DS:DX = <Ind. Buffer>**  
*Output:*    **CARRY flag = 1 (Errore)**    **CARRY flag = 0 (Nessun Errore)**  
               **AX = <Codice errore>**        **AX = <Numero di byte scritti>**

Questa funzione scrive sul file associato al numero di Handle riportato in **BX**; il registro **CX** specifica il numero di byte che si devono scrivere, mentre **DS:DX** puntano al buffer in cui sono collocati i dati che devono essere scritti. Conclusa l'operazione di scrittura, viene aggiornato il puntatore del file, in modo che punti dopo i dati appena scritti. Se l'operazione di scrittura ha successo, viene azzerato il **CARRY flag** e restituito, in **AX**, il numero di byte scritti; se questo valore è minore di quanto era stato scritto in **CX**, non vi è più spazio disponibile su disco, per completare l'operazione di scrittura. Se invece, si è verificato un errore, viene settato il **CARRY flag** e riportato, sempre in **AX**, il codice di errore, in particolare:

**AX = 05**       ->    Accesso inibito (file aperto solo in lettura)  
**AX = 06**       ->    Numero di handle non valido

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

**FUNZIONE 65: CANCELLA UN FILE**

*Input:*     **AH = 65 (41H)**  
               **DS:DX = <Ind. nome file in ASCIIZ>**  
*Output:*    **CARRY flag = 1 (Errore) CARRY flag = 0 (Nessun Errore)**  
               **AX = <Codice errore>**

Questa funzione cancella un file con nome e percorso indicati; questa stringa deve essere fornita in formato **ASCIIZ** (stringa ASCII terminata dal codice **0**). La coppia di registri **DS:DX** contiene l'indirizzo della stringa in questione.

Quando la funzione viene eseguita con successo, restituisce il **CARRY flag** azzerato; viceversa viene settato il flag e riportato in **AX** uno dei seguenti codici di errore:

**AX = 02**     -> File non trovato  
**AX = 03**     -> Percorso non trovato  
**AX = 05**     -> Accesso inibito (file con attributo di sola lettura)

**N.B.**

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

**FUNZIONE 66: SPOSTA IL PUNTATORE SU UN FILE**

*Input:*     **AH = 66 (42H)**  
               **AL = <Codice spostamento>**  
               **BX = <Handle>**  
               **CX:DX = <Offset puntatore>**  
*Output:*    **CARRY flag = 1 (Errore) CARRY flag = 0 (Nessun Errore)**  
               **AX = <Codice errore>**                **DX:AX = <Nuovo puntatore>**

Questa funzione cambia la posizione logica di lettura/scrittura in un file. Per richiamare questo servizio, si memorizza il numero di Handle associato al file, nel registro **BX**; quindi tramite i registri **CX** (word alta) e **DX** (word bassa), si riporta il numero intero, a 32 bit, che indica la nuova posizione del puntatore. Questo offset può avere tre riferimenti diversi, a seconda del valore del registro **AL**:

**AL = 0**     -> L'offset viene preso in riferimento all'inizio del file e il puntatore viene trasferito di **CX:DX** byte a partire da quel punto.  
**AL = 1**     -> L'offset viene riferito alla posizione attuale del puntatore del file.  
**AL = 2**     -> L'offset viene preso a partire dall'attuale fine del file.

Se la funzione viene eseguita con successo, azzerata il **CARRY flag** e restituisce, nella coppia di registri **DX:AX**, la nuova posizione del puntatore del file in relazione al suo inizio; viceversa viene settato il **CARRY flag** e riportato, sempre in **AX**, il codice di errore, in particolare:

**AX = 01**     -> Codice di spostamento, in **AL**, non valido.  
**AX = 06**     -> Numero di handle non valido.

Questa funzione può essere utilizzata in vari modi:

- 1) Per collocare il puntatore del file su una posizione arbitraria dello stesso, settare **AL=0** e specificare, in **CX:DX**, tale posizione in relazione all'inizio del file.
- 2) Per posizionare il puntatore alla fine del file settare **AL=2** e **CX:DX=0**.
- 3) Per determinare la posizione attuale del puntatore nel file, in relazione al suo inizio, settare **AL=1** e **CX:DX=0**.

#### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

### FUNZIONE 68: CONTROLLO DELL'I/O PER I DISPOSITIVI

*Input:*        **AH = 68 (44H)**  
                   **AL = <Numero della Subfunzione>**  
                   **[Altri registri dipendono dalla Subfunzione]**

*Output:*      **CARRY flag = 1        (Errore)    CARRY flag = 0    (Nessun Errore)**  
                   **AX = <Codice errore>                    [Altri registri dipendono dalla Subfunzione]**

Questa funzione si comporta in modo analogo alla funzione 68 (44H) del sistema operativo MS-DOS; l'utente può quindi consultare il manuale relativo, per quanto concerne le varie modalità di utilizzo ed i relativi registri gestiti di volta in volta.

#### N.B.

Questa funzione può comportare l'elaborazione di dati sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

### FUNZIONE 71: ACQUISIZIONE DELLA DIRECTORY

*Input:*        **AH = 71 (47H)**  
                   **DL = <Numero drive>**  
                   **DS:SI = <Ind. buffer 64 byte>**

*Output:*      **CARRY flag = 1        (Errore)    CARRY flag = 0    (Nessun Errore)**  
                   **AX = <Codice errore>                    DS:SI = <Directory in ASCIIZ>**

Viene restituita la directory corrente, relativa all'unità disco indicata nel registro **DL**, come indicato nella precedente funzione **54**: (**0**= drive corrente, **1**=drive **A** del P.C., **2**=drive **B** del P.C., **3**=drive **C** del P.C., ecc., **13 (0DH)**=RAM-Disk **M**). Nei registri **DS:DI**, deve essere riportato l'indirizzo di un buffer di 64 byte. La funzione, se eseguita correttamente, azzerà il **CARRY flag** e riempie il buffer con una stringa, in formato ASCII, che indica la directory corrente del drive indicato. Viceversa viene settato il flag e restituito il codice di errore **15 (0FH)** in **AX**.

#### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 76: TERMINA L'ESECUZIONE DEL PROCESSO

*Input:*     **AH = 76 (4CH)**

*Output:*    **Nessuno**

Come la funzione **0**, termina l'esecuzione del programma corrente, restituendo il controllo al **Monitor/Debugger 188**.

## FUNZIONE 78: TROVA LA PRIMA VOCE DI DIRECTORY

*Input:*     **AH = 78 (4EH)**

**CX = <Attributo>**

**DS:DX = <Ind. stringa ASCIIZ>**

*Output:*    **CARRY flag = 1        (Errore)    CARRY flag = 0    (Nessun Errore)**

**AX = <Codice errore>                   DTA = <Informazioni directory>**

Questa funzione esplora una directory per cercare il primo file con l'attributo ed il nome specificato. I registri **DS:DX** devono contenere l'indirizzo di una stringa, in formato **ASCIIZ**, che contiene il nome del percorso e del file da ricercare (nel caso di ricerca nei dischi del P.C. sono validi entrambi i caratteri jolly \* e ?, mentre il RAM-Disk é valida solo la dicitura \*.\*). Nel registro **CX**, inoltre, deve essere collocato il codice relativo agli attributi del file da ricercare (si veda la corrispondenza indicata nella funzione **60**).

### N.B.

Prima di chiamare questa funzione é necessario settare opportunamente un'area di **DTA**, di almeno **43 byte**, chiamando la funzione **26**, descritta in precedenza.

Se questa funzione trova la corrispondenza in una voce di directory con il nome specificato, azzerà il **CARRY flag** e riempie i primi **43 byte** della **DTA** con i seguenti dati:

**Byte 0÷20:**   Area usata dal sistema operativo **GDOS 188** per la funzione **79** (trova la prossima voce della directory).

**Byte 21:**     Attributi del file trovato.

**Byte 22÷23:**   Data del file trovato.

**Byte 24÷25:**   Ora del file trovato.

**Byte 26÷29:**   Dimensione del file in byte.

**Byte 30÷42:**   Nome ed estensione del file (stringa **ASCIIZ**).

Se la funzione, invece, non é stata eseguita con successo, viene settato il **CARRY flag** e restituito il codice di errore nel registro **AX**, in particolare:

**AX = 02**       ->   File non trovato.

**AX = 03**       ->   Percorso non trovato.

**AX = 18 (12H)** ->   Non ci sono più file; corrispondenza non trovata.

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota é collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 79: TROVA LA PROSSIMA VOCE DI DIRECTORY

*Input:*     **AH = 79 (4FH)**  
               **DTA = <Risultato della funzione 78>**  
*Output:*   **CARRY flag = 1**        (Errore)   **CARRY flag = 0**    (Nessun Errore)  
               **AX = <Codice errore>**                    **DTA = <Informazioni directory>**

Questa funzione continua l'esplorazione della directory, con un nome che potrebbe trovare corrispondenza con più di una voce perché contiene caratteri jolly. Quando si chiama questa procedura, la **DTA** deve contenere i dati restituiti da una precedente chiamata alla funzione **78** o **79**. Se viene trovata una voce di directory che corrisponde, viene azzerato il **CARRY flag** e aggiornata opportunamente la **DTA**. Viceversa viene settato il flag e restituito in **AX**, il codice di errore **18 (12H)**: non ci sono più file.

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

## FUNZIONE 86: CAMBIA IL NOME UN FILE

*Input:*     **AH = 86 (56H)**  
               **DS:DX = <Ind. nome file in ASCIIZ>**  
               **ES:DI = <Ind. nuovo nome in ASCIIZ>**  
*Output:*   **CARRY flag = 1**        (Errore)   **CARRY flag = 0**    (Nessun Errore)  
               **AX = <Codice errore>**

Questa funzione cambia il nome di un file; può, però, anche spostare un file, da una directory ad un'altra. Il file in quanto tale non viene spostato, soltanto la voce della directory, il che significa che il percorso vecchio e quello nuovo, debbono essere sullo stesso lettore.

Le specifiche del file vengono fornite sotto forma di stringhe **ASCIIZ**. La coppia di registri **DS:DX** punta alla stringa con il vecchio nome ed **ES:DI** puntano alla stringa di quello nuovo.

Quando la funzione viene eseguita con successo, restituisce il **CARRY flag** azzerato; viceversa viene settato il flag e riportato in **AX** uno dei seguenti codici di errore:

**AX = 02**        -> File non trovato  
**AX = 03**        -> Percorso non trovato  
**AX = 05**        -> Accesso inibito (file con attributo di sola lettura)  
**AX = 17 (11H)** -> Non stanno sullo stesso dispositivo

### N.B.

Questa funzione può comportare la manipolazione di file sul P.C.; quindi sarà eseguita correttamente, solo se la scheda remota è collegata, tramite la linea di console, all'emulazione terminale intelligente del programma **GET188**.

**TABELLA RIASSUNTIVA DELLE FUNZIONI DELL'INTERRUPT 21H**

Di seguito vengono riportate le tabelle riassuntive delle funzioni relative al vettore di interrupt **21H**.

FUNZIONE	INPUT	OUTPUT
<b>Termina esecuzione Programma</b>	AH = 0	----
<b>Input da linea Ausiliaria</b>	AH = 3	AL = <Dato Ricevuto>
<b>Output su linea Ausiliaria</b>	AH = 4 AL = <Dato da trasmettere>	----
<b>Output su stampante locale</b>	AH = 5 AL = <Dato da trasmettere>	----
<b>Svuota il buffer dei file</b>	AH = 13 (0DH)	----
<b>Seleziona il drive corrente</b>	AH = 14 (0EH) DL = <Numero del drive>	AL = <Num. unità installate>
<b>Acquisizione drive corrente</b>	AH = 25 (19H)	AL = <Drive corrente>
<b>Setta indirizzo della DTA</b>	AH = 26 (1AH) DS:DX = <Indirizzo DTA>	----
<b>Acquisizione della Data</b>	AH = 42 (2AH)	AL = <Giorno settimana> CH = 0 CL = <Anno> DH = <Mese> DL = <Giorno>
<b>Settaggio della Data</b>	AH = 43 (2BH) CL = <Anno> DH = <Mese> DL = <Giorno>	AL = 0
<b>Acquisizione dell'Ora</b>	AH = 44 (2CH)	CH = <Ora> CL = <Minuti> DH = <Secondi>
<b>Settaggio dell'Ora</b>	AH = 45 (2DH) CH = <Ora> CL = <Minuti> DH = <Secondi>	AL = 0
<b>Acquisizione del numero di versione</b>	AH = 48 (30H)	AX = <Numero versione> BX:CX = <Numero di serie>
<b>Acquisizione dello spazio libero su disco</b>	AH = 54 (36H) DL = <Numero drive>	AX = <Settori per Cluster> BX = <Cluster non utilizzati> CX = <Byte per settore> DX = <Tot. Cluster su disco> oppure: AX = FFFFH (Drive errato)

**FIGURA 32: TABELLA 1 DELLE FUNZIONI DELL'INTERRUPT 21H**

FUNZIONE	INPUT	OUTPUT
<b>Make di un file</b>	AH = 60 (3CH) CX = <Attributo> DS:DX = <Ind. nome file in ASCIIZ>	CARRY flag = 0 AX = <Handle> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Open di un file</b>	AH = 61 (3DH) CX = <Codice accesso> DS:DX = <Ind. nome file in ASCIIZ>	CARRY flag = 0 AX = <Handle> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Close di un file</b>	AH = 62 (3EH) BX = <Handle>	CARRY flag = 0 oppure: CARRY flag = 1 AX = <Codice errore>
<b>Legge da un file</b>	AH = 63 (3FH) BX = <Handle> CX = <Num. byte da leggere> DS:DX = <Ind. buffer>	CARRY flag = 0 AX = <Num. byte letti> DS:DX = <Ind. buffer > oppure: CARRY flag = 1 AX = <Codice errore>
<b>Scrive su un file</b>	AH = 64 (40H) BX = <Handle> CX = <Num. byte da scrivere> DS:DX = <Ind. buffer>	CARRY flag = 0 AX = <Num. byte scritti> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Cancella un file</b>	AH = 65 (41H) DS:DX = <Ind. nome file in ASCIIZ>	CARRY flag = 0 oppure: CARRY flag = 1 AX = <Codice errore>
<b>Sposta il puntatore su un file</b>	AH = 66 (42H) AL = <Codice spostamento> BX = <Handle> CX:DX = <Offset puntatore>	CARRY flag = 0 DX:AX = <Nuovo puntatore> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Controllo dell'I/O per i dispositivi</b>	AH = 68 (44H) AL = <Numero subfunzione> [Altri registri dipendono dalla Subf.]	CARRY flag = 0 [Altri reg. dipendono dalla Subf.] oppure: CARRY flag = 1 AX = <Codice errore>

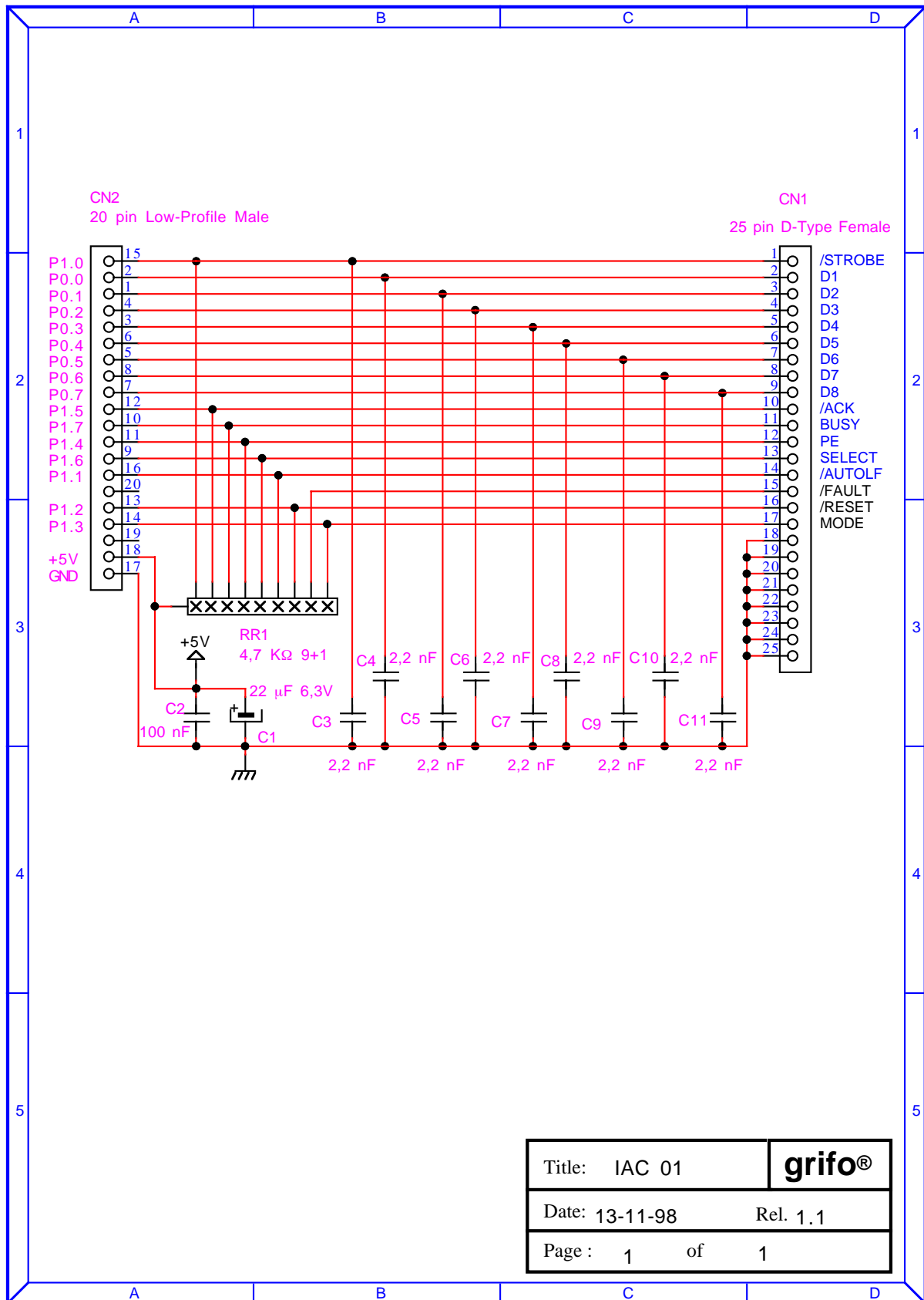
FIGURA 33: TABELLA 2 DELLE FUNZIONI DELL'INTERRUPT 21H

FUNZIONE	INPUT	OUTPUT
<b>Acquisizione della directory</b>	AH = 71 (47H) AL = <Numero drive> DS:SI = <Ind. buffer 64 byte>	CARRY flag = 0 DS:SI= <Dir. in ASCIIZ> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Termina esecuzione del processo</b>	AH = 76 (4CH)	----
<b>Trova la prima voce di directory</b>	AH = 78 (3EH) CX = <Attributo> DS:DX = <Ind. stringa in ASCIIZ>	CARRY flag = 0 DTA= <Info. directory> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Trova la prossima voce di directory</b>	AH = 79 (3FH)	CARRY flag = 0 DTA= <Info. directory> oppure: CARRY flag = 1 AX = <Codice errore>
<b>Cambia il nome di un file</b>	AH = 78 (3EH) DS:DX = <Ind. nome file in ASCIIZ> ES:DI = <Ind. nuovo nome in ASCIIZ>	CARRY flag = 0 oppure: CARRY flag = 1 AX = <Codice errore>

**FIGURA 34: TABELLA 3 DELLE FUNZIONI DELL'INTERRUPT 21H**



APPENDICE A: SCHEMI INTERFACCIE I/O DIGITALI



Title: IAC 01	grifo®
Date: 13-11-98	Rel. 1.1
Page : 1	of 1

FIGURA A1: SCHEMA ELETTRICO DELLA SCHEDA IAC 01



## APPENDICE B: INDICE ANALITICO

**Simboli**

80C188 2, 3, 38, 50, 63

**A**

ABILITA LA FORMATTAZIONE DELLA RAM-DISK, Funzione 31 INT 3BH 76

ACQUISIZIONE DEL DRIVE CORRENTE, Funzione 25 INT 21H 82

ACQUISIZIONE DEL NUMERO DI VERSIONE, Funzione 48 INT 21H 84

ACQUISIZIONE DELLA DATA, Funzione 42 INT 21H 83

ACQUISIZIONE DELLA DIRECTORY, Funzione 71 INT 21H 89

ACQUISIZIONE DELLO SPAZIO LIBERO SU DISCO, Funzione 54 INT 21H 84

ACQUISIZIONE DELL'ORA, Funzione 44 INT 21H 83

ACQUISIZIONE DI UN TASTO, Funzione 0 INT 3BH 72

ALTA LUMINOSITA', Funzione 25 INT 3BH 75

ASSEMBLATORE IN LINEA, Comando Monitor/Debugger 188 58

AUTORUN 25, 49

**C**

CAMBIA IL NOME UN FILE, Funzione 86 INT 21H 91

CANCELLA UN FILE, Funzione 65 INT 21H 88

CANCELLAZIONE DI FILE DA RAM-DISK, Comando TOOLS 188 69

CAVO DI COMUNICAZIONE SERIALE 4

CLEAR SCREEN, Funzione 21 INT 3BH 74

CLEAR END OF LINE, Funzione 22 INT 3BH 74

CLOCK 25

CLOSE DI UN FILE, Funzione 62 INT 21H 86

COME INIZIARE 8

CONSOLE 3, 4, 14, 24, 29, 73

CONTROLLO DELLA RAM-DISK, Funzione 33 INT 3BH 77

CONTROLLO DELL'I/O PER I DISPOSITIVI, Funzione 68 INT 21H 89

COPIA DI FILE, Comando TOOLS 188 68

COPIA DI UN BLOCCO DI MEMORIA, Comando Monitor/Debugger 188 59

CPU 2, 3, 38, 50, 63

**D**

DEASSEMBLA UN BLOCCO DI MEMORIA, Comando Monitor/Debugger 188 62

DEB 01 3, 27

DEBUG 28

DELETE LINE, Funzione 23 INT 3BH 75

DISCO DISTRIBUITO 7

**E**

EEPROM 3, 25, 26, 77, 78

EPROM 3, 4, 39

EPROM DI LAVORO 39

EPROM UTENTE 22, 40

## segue E

ESECUZIONE PROGRAMMA ALL'INDIRIZZO E0000H, Comando Monitor/Debugger 188	59
ESECUZIONE PROGRAMMA ALL'INDIRIZZO F0000H, Comando Monitor/Debugger 188	62
ESECUZIONE DI UN BLOCCO DI PROGRAMMA, Comando Monitor/Debugger 188	61
ESECUZIONE DI UNA O PIU' ISTRUZIONI, Comando Monitor/Debugger 188	64
ESECUZIONE DI UNA PROCEDURA, Comando Monitor/Debugger 188	63

## F

FIRMWARE DI PROGRAMMAZIONE	6
FLAG DI CONSOLE	14, 38
FLASH EPROM	3, 4, 25, 41
FLASH EPROM DI DEBUG	42
FLASH EPROM DI LAVORO	41
FLASH EPROM UTENTE	43
FORMATTAZIONE DELLA RAM-DISK, Comando TOOLS 188	69
FORMATTAZIONE DELLA RAM-DISK, Funzione 32 INT 3BH	76

## G

<b>GDOS 188</b>	1, 2, 6, 7, 24
CARATTERISTICHE TECNICHE	24
FUNZIONI DEL SISTEMA OPERATIVO	71
JUMPER DI SETTAGGIO	28
VERSIONI	25
GENERALITA'	2
GESTIONE DELLA FLASH EPROM	45
GESTIONE DELLA RAM-DISK	48
<b>GET188</b>	1, 7, 10, 25
CARATTERISTICHE GENERALI	10
CODICI DEI TASTI SPECIALI	19
CREAZIONE DI UN'IMMAGINE BINARIA DELLA EPROM UTENTE	22
DESCRIZIONE	11
EDITOR	12
EMULAZIONE TERMINALE	14
ESECUZIONE DI UN COMPILATORE	20
INSTALLAZIONE	11
MATERIALE NECESSARIO	10
SEQUENZE DI CONTROLLO	17
<b>GPC®</b>	2, 3, 25
<b>GPC® 188D</b>	
BAUD RATE DELLA LINEA DI CONSOLE	29
CONFIGURAZIONE DELLA RAM	29
MAPPAGGIO DELLE PERIFERICHE	50
ORGANIZZAZIONE DELLA MEMORIA	30
<b>GPC® 188F</b>	
BAUD RATE DELLA LINEA DI CONSOLE	29
CONFIGURAZIONE DELLA RAM	29
MAPPAGGIO DELLE PERIFERICHE	52
ORGANIZZAZIONE DELLA MEMORIA	32

**segue G****GPC® 883**

MAPPAGGIO DELLE PERIFERICHE 54  
ORGANIZZAZIONE DELLA MEMORIA 33

**GPC® 884**

MAPPAGGIO DELLE PERIFERICHE 56  
ORGANIZZAZIONE DELLA MEMORIA 35

**I**

I/O TTL 3, 24, 27, 81, A-1

IAC 01 3, 27, A-1

INPUT DA CONSOLE, Funzione 2 INT 3BH 73

INPUT DALLA LINEA AUSILIARIA, Funzione 3 INT 21H 80

INPUT DI UN BYTE, Comando Monitor/Debugger 188 61

INPUT DI UNA WORD, Comando Monitor/Debugger 188 60

INSERT LINE, Funzione 24 INT 3BH 75

INTERFACCIE PER I/O DIGITALI 27, A-1

INTERRUPT 21H 80

TABELLA RIASSUNTIVA DELLE FUNZIONI 92

INTERRUPT 3BH 71

TABELLA RIASSUNTIVA DELLE FUNZIONI 79

INTRODUZIONE 1

**J**

JUMPER

CONFIGURAZIONE DELLA RAM 29

RUN/DEBUG 28

SELEZIONE DEL BAUD RATE 29

**L**

LEGGE DA UN FILE, Funzione 63 INT 21H 87

LEGGIMI.ITA 2

LETTURA DI UN BYTE DALL'EEPROM, Funzione 34 INT 3BH 77

LINEA AUSILIARIA 3, 24, 73, 80, 81

LINEA DI CONSOLE 3, 4, 14, 24, 29, 73

LINEA SERIALE A 3, 4, 14, 24, 29, 73

LINEA SERIALE B 3, 24, 73, 80, 81

LUMINOSITA' NORMALE, Funzione 26 INT 3BH 76

**M**

MAKE DI UN FILE, Funzione 60 INT 21H 85

MANUALE UTENTE GDOS 188 6

MAPPAGGIO DELLE PERIFERICHE 50

MATERIALE NECESSARIO 3

MEMORIA 30

MODALITA' RUN / DEBUG 28

**segue M**

MONITOR / DEBUGGER 188 2, 6, 7  
COMANDI 58  
DESCRIZIONE 57  
FORMATO DEI PARAMETRI 57  
TABELLA RIASSUNTIVA DEI COMANDI 66  
VETTORI DI INTERRUPT 58

**O**

OPEN DI UN FILE, Funzione 61 INT 21H 86  
ORGANIZZAZIONE DELLA EPROM 39  
ORGANIZZAZIONE DELLA FLASH EPROM 41  
ORGANIZZAZIONE DELLA MEMORIA 30  
OUTPUT DI UN BYTE, Comando Monitor/Debugger 188 62  
OUTPUT DI UNA WORD, Comando Monitor/Debugger 188 65  
OUTPUT SU CONSOLE DI "BELL", Funzione 15 INT 3BH 73  
OUTPUT SU CONSOLE DI "CR+LF", Funzione 16 INT 3BH 73  
OUTPUT SU CONSOLE, Funzione 3 INT 3BH 73  
OUTPUT SU LINEA AUSILIARIA, Funzione 4 INT 21H 81  
OUTPUT SU STAMPANTE LOCALE, Funzione 5 INT 21H 81

**P**

PASCAL 188 2, 6, 7, 9, 16, 39, 47, 59  
PBA 38, 50, 71  
PERSONAL COMPUTER 4  
POSIZIONAMENTO DEL CURSORE, Funzione 20 INT 3BH 74  
PROGRAMMA IN AUTORUN 49

**R**

RAM DI LAVORO 29, 30, 37  
RAM-DISK 3, 24, 29, 30, 48, 49, 69, 76, 77  
REAL TIME CLOCK 3, 24, 26, 69, 83, 84  
REGISTRO PBA 38, 50, 71  
RIAVVIA IL GDOS 188, Comando Monitor/Debugger 188 63  
RICEZIONE DI UN FILE, Comando Monitor/Debugger 188 65  
RIEMPIMENTO DI UN BLOCCO DI MEMORIA, Comando Monitor 188 60  
RUN 28

**S**

SCHEDA DI CONTROLLO 2, 3, 25  
SCHEDA REMOTA 2, 3, 25  
SCRITTURA DI UN BYTE NELL'EEPROM, Funzione 35 INT 3BH 78  
SCRITTURA NELLA RAM PROTETTA IN SCRITTURA, Funzione 36 INT 3BH 78  
SCRIVE SU UN FILE, Funzione 64 INT 21H 87  
SELEZIONE DEL DRIVE CORRENTE, Funzione 14 INT 21H 82  
SELEZIONE DEL MODO OPERATIVO 28  
SELEZIONE DRIVE E VISUALIZZAZIONE DIRECTORY, Comando TOOLS 188 68  
SETTA L'INDIRIZZO DELLA DTA, Funzione 26 INT 21H 82

## segue S

SETTAGGIO DEL REAL TIME CLOCK, Comando TOOLS 188 69  
SETTAGGIO DELLA DATA, Funzione 43 INT 21H 83  
SETTAGGIO DELL'ORA, Funzione 45 INT 21H 84  
SOFTWARE DI LAVORO 5  
SOSTITUZIONE DI DATI IN MEMORIA, Comando Monitor/D 63  
SPOSTA IL PUNTATORE SU UN FILE, Funzione 66 INT 21 88  
SRAM 3, 24, 29, 30, 78  
STAMPANTE LOCALE 3, 24, 27, 81, A-1  
STATO DELLA CONSOLE, Funzione 1 INT 3BH 72  
STATO DELLA LINEA AUSILIARIA, Funzione 4 INT 3BH 73  
SVUOTA I BUFFER DEI FILE, Funzione 13 INT 21H 81

## T

TERMINA L'ESECUZIONE DEL PROCESSO, Funzione 76 INT 21H 90  
TERMINA L'ESECUZIONE DEL PROGRAMMA, Funzione 0 INT 21H 80  
TOOLS 188 2, 6, 7, 39, 47, 62, 67  
  COMANDI 68  
  DESCRIZIONE 67  
  ESECUZIONE DEL PROGRAMMA 67  
TROVA LA PRIMA VOCE DI DIRECTORY, Funzione 78 INT 21H 90  
TROVA LA PROSSIMA VOCE DI DIRECTORY, Funzione 79 INT 21H 91

## U

ULTIMO SEGMENTO DI TPA 38  
USCITA DAL PROGRAMMA, Comando TOOLS 188 70

## V

VARIAZIONE DEL NOME DI UN FILE SU RAM-DISK, Comando TOOLS 188 69  
VERSIONI 1, 25  
VETTORE DI INTERRUPT 21H 80  
VETTORE DI INTERRUPT 3BH 71  
VISUALIZZAZIONE DELL'ELENCO DEI COMANDI, Comando Monitor/Debugger 188 61  
VISUALIZZAZIONE DI UN BLOCCO DI MEMORIA, Comando Monitor/Debugger 188 60  
VISUALIZZAZIONE IN FORMATO HEX INTEL, Comando Monitor/Debugger 188 65  
VISUALIZZAZIONE O MODIFICA DEI REGISTRI, Comando Monitor/Debugger 188 63

## W

WATCH DOG 24, 25, 26  
WFLASH 7, 45  
  CARICAMENTO ED ESECUZIONE 45  
  CONFIGURAZIONE SCHEDA 45  
  FASI DEL PROGRAMMA DI GESTIONE 46

