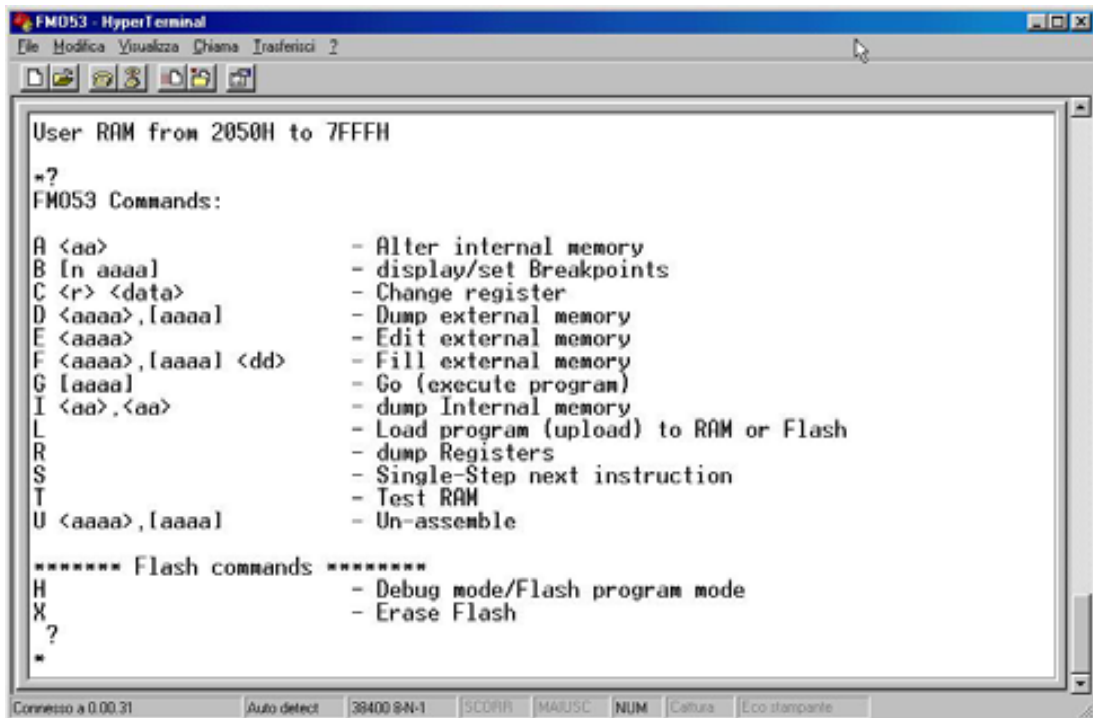


FM053

Flash Monitor Debugger per famiglia 51

MANUALE UTENTE



```
FM053 - HyperTerminal
File Modifica Visualizza Chiama Trasferisci 2
[Icons]
User RAM from 2050H to 7FFFH
*?
FM053 Commands:
A <aa> - Alter internal memory
B [n aaaa] - display/set Breakpoints
C <r> <data> - Change register
D <aaaa>,[aaaa] - Dump external memory
E <aaaa> - Edit external memory
F <aaaa>,[aaaa] <dd> - Fill external memory
G [aaaa] - Go (execute program)
I <aa>,<aa> - dump Internal memory
L - Load program (upload) to RAM or Flash
R - dump Registers
S - Single-Step next instruction
T - Test RAM
U <aaaa>,[aaaa] - Un-assemble

***** Flash commands *****
H - Debug mode/Flash program mode
X - Erase Flash
?
*
```

grifo[®]

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

<http://www.grifo.com>

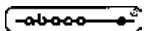
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

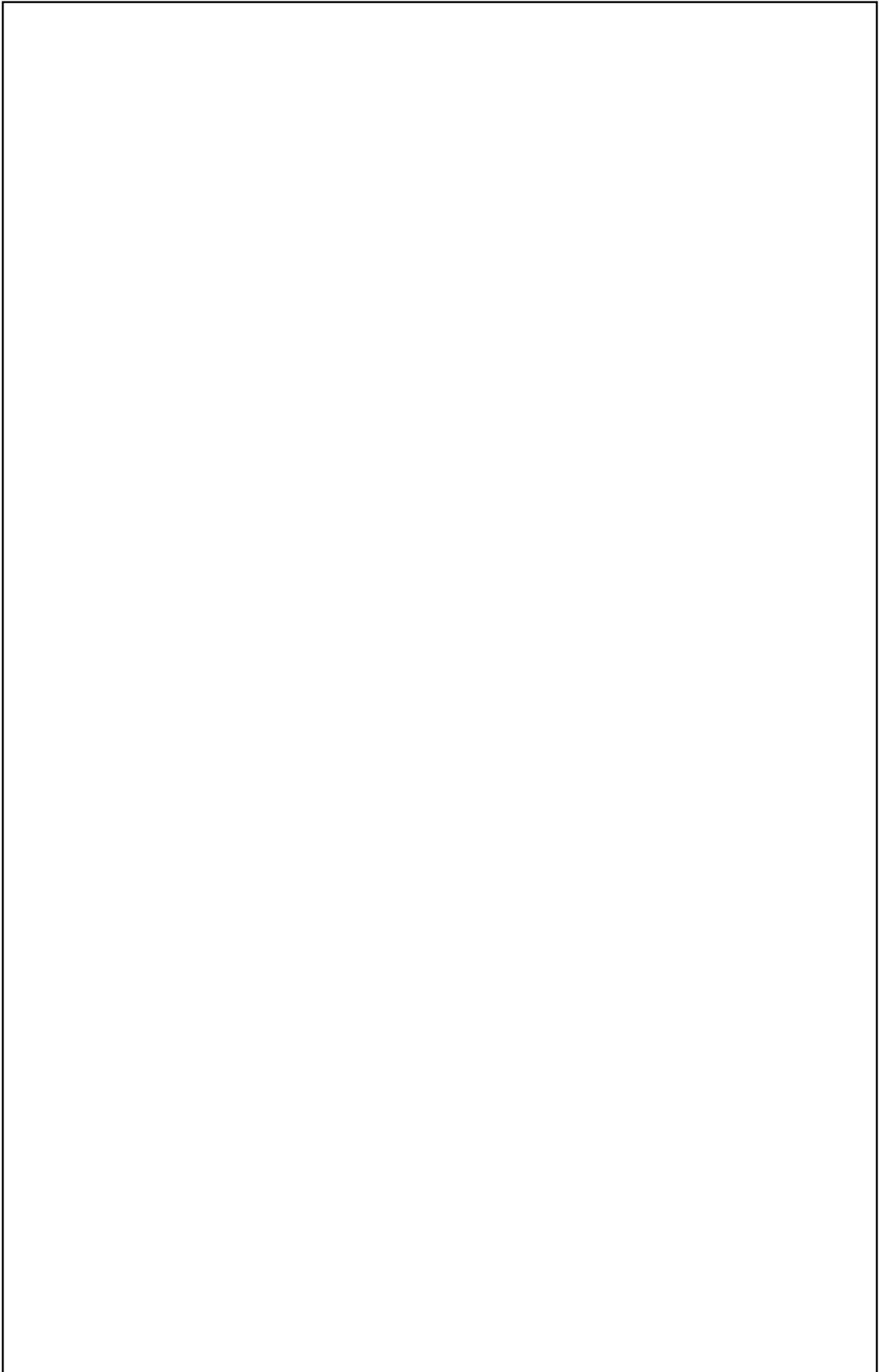


FM053

Rel. 3.00

Edizione 13 Giugno 2006

, GPC[®], grifo[®], sono marchi registrati della ditta grifo[®]



FMO53

Flash Monitor Debugger per famiglia 51

MANUALE UTENTE

FMO53 e' un pacchetto software disponibile per molte delle schede **grifo**[®] dotate di microprocessori appartenenti alla famiglia **Intel 51**. E' interattivo, veloce, di facile utilizzo e comprende un debugger, un monitor in linguaggio macchina ed un gestore di **FLASH EPROM**, che permettono di scaricare, provare e salvare il programma applicativo sviluppato dall'utente.

FMO53 richiede e sfrutta pochissime risorse dell'hardware utilizzato e si può utilizzare con l'ausilio di un semplice **PC**, collegato in seriale **RS 232**. Al fine di non usare risorse necessarie all'applicazione da sviluppare, sulle schede provviste di due linee seriali é possibile selezionare quella usata dall'**FMO53**.

Include interessanti funzioni come: visualizzazione ed alterazione del contenuto delle memorie; gestione di punti d'arresto; esecuzione codice a singola istruzione od a velocità reale; un completo disassemblatore; visualizzazione e modifica dei registri; test della memoria **SRAM**; scaricamento di un programma utente; cancellazione e programmazione della **FLASH**; lancio automatico di un programma salvato; ecc.

L'**FMO53** può essere usato in abbinamento alla maggioranza degli ambienti di sviluppo disponibili sempre per la famiglia **I51** (Compilatori C, BASIC, PASCAL, ecc.) con cui si integra ottenendo un unico ambiente di sviluppo e prova.

grifo[®]

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

<http://www.grifo.com>

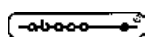
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



FMO53

Rel. 3.00

Edizione 13 Giugno 2006



, **GPC**[®], **grifo**[®], sono marchi registrati della ditta **grifo**[®]

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 sul 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

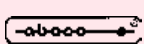


Attenzione: Pericolo di alta tensione



Attenzione: Dispositivo sensibile alle cariche elettrostatiche

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
VERSIONE	3
INFORMAZIONI GENERALI	4
MATERIALE NECESSARIO	7
SCHEDA DI CONTROLLO	7
PERSONAL COMPUTER	7
CAVO DI COMUNICAZIONE SERIALE	8
SOFTWARE DI LAVORO	9
AMBIENTE DI SVILUPPO PROGRAMMA APPLICATIVO	10
PROGRAMMA EMULAZIONE TERMINALE SERIALE	12
PROGRAMMATORE DI EPROM	12
DESCRIZIONE FMO53	13
CONFIGURAZIONE SCHEDA	13
RISORSE UTILIZZATE	18
COMUNICAZIONE SERIALE	19
CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE	20
GESTIONE FLASH EPROM	22
CODICE SU FLASH	22
DATI SU FLASH	23
SELEZIONE MODO OPERATIVO	24
INTERRUPTS	24
INTEGRAZIONE CON AMBIENTE DI SVILUPPO	25
UTILIZZO CON BASCOM 8051	25
CONFIGURAZIONI PER COMUNICAZIONE SERIALE	25
DIRETTIVE DEL PROGRAMMA APPLICATIVO	27
UTILIZZO CON GET51	29
CONFIGURAZIONI PER COMUNICAZIONE SERIALE	29
UTILIZZO CON HYPERTERMINAL	31
CONFIGURAZIONI PER COMUNICAZIONE SERIALE	31
UTILIZZO CON μ C/51	34
CONFIGURAZIONI PER COMUNICAZIONE SERIALE	35
DIRETTIVE DEL PROGRAMMA APPLICATIVO	35
COMANDI	37
ALTERA MEMORIA INTERNA	38
SETTA E VISUALIZZA PUNTI D'ARRESTO	38
MODIFICA REGISTRI	39
VISUALIZZA MEMORIA DATI ESTERNA	39
MODIFICA MEMORIA DATI ESTERNA	39
RIEMPE MEMORIA DATI ESTERNA	40
ESEGUE A VELOCITÀ REALE	40

ABILITA COMANDI PER FLASH	40
VISUALIZZA MEMORIA INTERNA	40
CARICA FILE	41
VISUALIZZA REGISTRI	42
ESEGUE A SINGOLO PASSO	42
TESTA SRAM DELLA SCHEDA	42
DISASSEMBLA CODICE	43
CANCELLA FLASH.....	43
VISUALIZZA COMANDI.....	43
COME INIZIARE	44
APPENDICE A: INDICE ANALITICO	A-1

INDICE DELLE FIGURE

FIGURA 1: COLLEGAMENTO SERIALE TRA PC DI SVILUPPO E SCHEDE DI CONTROLLO	8
FIGURA 2: CONNETTORE ED ACCESSORI PER COLLEGAMENTO SERIALE	9
FIGURA 3: MODALITÀ DI LAVORO	11
FIGURA 4: CONFIGURAZIONE SCHEDE (1 DI 2)	14
FIGURA 5: CONFIGURAZIONE SCHEDE (2 DI 2)	15
FIGURA 6: ORGANIZZAZIONE MEMORIE IN MODALITÀ DEBUG	16
FIGURA 7: ORGANIZZAZIONE MEMORIE IN MODALITÀ AUTORUN	17
FIGURA 8: SELEZIONE LINEA SERIALE USATA	19
FIGURA 9: INDIRIZZI AREE DI MEMORIA IN MODALITÀ DEBUG	20
FIGURA 10: INDIRIZZI AREE DI MEMORIA IN MODALITÀ AUTORUN	21
FIGURA 11: DIMENSIONI MASSIME AREE DI MEMORIA	21
FIGURA 12: CONFIGURAZIONE EMULAZIONE TERMINALE BASCOM 8051	26
FIGURA 13: CONFIGURAZIONE SCARICAMENTO FILE CON BASCOM 8051	26
FIGURA 14: SCARICAMENTO FILE CON BASCOM 8051	27
FIGURA 15: CONFIGURAZIONE SERIALE DEL GET51	29
FIGURA 16: CONFIGURAZIONE TIPO TERMINALE CON GET51	30
FIGURA 17: SCARICAMENTO FILE CON GET51	30
FIGURA 18: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (1 DI 4)	31
FIGURA 19: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (2 DI 4)	32
FIGURA 20: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (3 DI 4)	32
FIGURA 21: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (4 DI 4)	33
FIGURA 22: SCARICAMENTO FILE CON HYPERTERMINAL (1 DI 2)	33
FIGURA 23: SCARICAMENTO FILE CON HYPERTERMINAL (2 DI 2)	34
FIGURA 24: ELENCO COMANDI	37
FIGURA 25: COMANDO ABILITAZIONE COMANDI PER FLASH	41
FIGURA 26: MESSAGGIO DI PRESENTAZIONE FMO53 SU SERIALE A	45
FIGURA 27: MESSAGGIO DI PRESENTAZIONE FMO53 SU SERIALE B	45
FIGURA 28: SCARICAMENTO ED ESECUZIONE DI UN PROGRAMMA IN SRAM	46
FIGURA 29: SCARICAMENTO E PROGRAMMAZIONE DI UN PROGRAMMA IN FLASH	47
FIGURA 30: PARTENZA AUTOMATICA PROGRAMMA IN AUTORUN	48



INTRODUZIONE

L'uso di questi dispositivi è rivolto - **IN VIA ESCLUSIVA** - a personale specializzato.

Questo prodotto non è un **componente di sicurezza** così come definito dalla direttiva **98-37/CE**.



I pin della/e scheda/e non sono dotati di protezione contro le cariche elettrostatiche. Visto che esiste un collegamento diretto tra numerosi pin delle schede ed i rispettivi pin dei componenti di bordo, e che quest'ultimi sono sensibili ai fenomeni ESD, il personale che maneggia la/e scheda/e è invitato a prendere tutte le precauzioni necessarie per evitare i possibili danni che potrebbero derivare dalle cariche elettrostatiche.

Scopo di questo manuale è 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.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall'**ASSISTENZA TECNICA AUTORIZZATA**, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

I dispositivi non possono essere utilizzati all'aperto. Si deve sempre provvedere ad inserire i moduli all'interno di un contenitore a norme di sicurezza che rispetti le vigenti normative. La protezione di questo contenitore non si deve limitare ai soli agenti atmosferici, bensì anche a quelli meccanici, elettrici, magnetici, ecc.

Per un corretto rapporto coi prodotti, è 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 tali dispositivi, è 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 problemi siano risolti. La **grifo®** non è inoltre responsabile dei problemi causati dalle modifiche dell'hardware dei calcolatori o dei sistemi operativi che si possono verificare nel tempo.

grifo® si riserva il diritto di apportare cambiamenti e/o miglioramenti ai prodotti descritti in questo manuale, ed allo stesso manuale, in qualunque momento senza darne notizia.

Il programma descritto in questo manuale è coperto da diritti d'autore. Né il programma né alcuna sua parte possono essere analizzati, disassemblati o modificati in alcun modo, con qualunque mezzo, per qualunque scopo.

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

VERSIONE

Il presente manuale è riferito alla versione **2.0** del programma **FMO53** ed alle eventuali versioni successive. La validità delle informazioni riportate è quindi subordinata al numero di versione in uso e l'utente deve sempre verificarne la giusta corrispondenza. Il numero di versione è chiaramente riportato sull'etichetta applicata alla EPROM in cui l'**FMO53** viene fornito e viene inoltre rappresentato nella sua videata di presentazione.

Normalmente l'**FMO53** viene sempre fornito con l'ultima versione disponibile, ma in caso di specifiche esigenze l'utente può richiedere anche una versione diversa, specificandolo in fase di ordine.

In questo manuale sono presenti delle informazioni relative ad altri programmi che costituiscono una parte integrante dell'**FMO53**: ognuno di questi ha il proprio numero di versione che, quando necessario, viene presentato in questo manuale. Infine anche l'hardware è dotato di una propria versione come indicato nei relativi manuali tecnici.

In caso di necessità di assistenza tecnica è di fondamentale importanza che l'utente, oltre alla descrizione del problema, fornisca i numeri di versione dei prodotti in uso.

Come ogni prodotto, anche l'**FMO53** è 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 ha subito, a seconda del numero di versione

FMO52 Ver. 1.0 -> Versione di sviluppo e prove interne.

FMO52 Ver. 1.1 -> Prima versione rilasciata.

FMO52 Ver. 1.2 -> Prevista esecuzione su schede dotate di microprocessore DALLAS, con le sue ridotte tempistiche di esecuzione; migliorata gestione della singola istruzione.

FMO52 Ver. 1.3 -> Aggiunta esecuzione su **GPC® 550**.

FMO53 Ver. 2.0 -> Aggiunta selezione seriale usata sulle schede provviste di due linee; aumentato baud rate di comunicazione seriale; eliminata inizializzazione seriale in caso di programma in autorun; aumentata dimensione massima del programma utente memorizzabile su FLASH; migliorata e velocizzata la gestione della FLASH; migliorato test della SRAM; corretto settaggio iniziale del program counter; variate modalità di pausa, ripresa ed interruzione delle rappresentazioni lunghe.

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

INFORMAZIONI GENERALI

FMO53 e' un pacchetto software interattivo, composto da un monitor debugger ed un gestore di FLASH EPROM, che consente di sviluppare l'applicazione dell'utente in modo veloce e confortevole. E' disponibile per la maggioranza delle schede **GPC®** appartenenti al carteggio industriale **grifo®**, dotate di microprocessori Intel 51. Grazie all'**FMO53** ogni utente e' in grado di mettere a punto il firmware per la scheda di controllo usata, senza dover ricorrere a costosi e complicati sistemi di sviluppo esterni, con un conseguente risparmio di tempo e denaro. L'economicità d'uso dell'**FMO53** e' confermata anche dall'esiguo materiale necessario per poterlo usare: e' sufficiente un normale PC collegato in seriale RS 232 alla scheda prescelta.

Il monitor debugger opera in linguaggio macchina e può quindi essere usato in abbinamento a qualsiasi linguaggio di programmazione, sempre per la famiglia I51; tra questi si possono ricordare numerosi assembleri, compilatori C, compilatori BASIC, compilatori PASCAL, ecc.

L'**FMO53** prevede numerose funzioni come visualizzazione ed alterazione del contenuto delle memorie, gestione di punti d'arresto, esecuzione codice a singola istruzione, esecuzione codice a velocità reale, ispezione e modifica dei registri del microcontrollore, un completo disassemblatore, test della RAM, cancellazione della FLASH, programmazione della FLASH con il programma utente e lancio automatico del programma salvato in FLASH. Tali funzioni sono quelle tipiche di un "In Circuit Emulator" hardware di cui l'**FMO53** costituisce un valido sostituto, realizzato completamente via software.

In termini operativi l'utente si deve limitare a scrivere il programma applicativo per la scheda usata, con il linguaggio di programmazione scelto; a questo punto deve trasformare il sorgente in codice eseguibile (compilarlo e/o assemblarlo) e quindi scaricarlo alla scheda tramite l'**FMO53**. Il programma, una volta scaricato, può essere eseguito sulla stessa scheda in modo da verificarne il funzionamento direttamente nel sistema reale; nel caso in cui il funzionamento manifesti alcuni problemi, l'utente li deve eliminare tutti ripetendo ciclicamente i passi elencati fino ad ora. Una volta ottenuto un funzionamento completamente corretto si può proseguire con il salvataggio del programma applicativo definitivo nella FLASH EPROM, sempre tramite l'**FMO53**. A questo punto la scheda e' pronta per essere installata sull'impianto finale in quanto alle successive accensioni il programma applicativo partirà in modo automatico. Qualora a distanza di tempo l'utente debba intervenire sul programma applicativo per verificarlo od aggiornarlo, l'**FMO53** può interrompere l'esecuzione del programma salvato e consentire tutte le operazioni precedentemente descritte, con il semplice spostamento di un apposito selettore a bordo scheda.

E' importante ricordare che l'**FMO53** usa pochissime risorse della scheda che lo esegue in modo da non essere intrusivo per il programma applicativo dell'utente. Comunque tali risorse (dettagliatamente elencate nella successiva descrizione) vengono completamente liberate quando si esegue il programma salvato in FLASH; l'utente può quindi lavorare come se il suo programma fosse da solo.

Le caratteristiche fondamentali dell'**FMO53** possono essere riassunte nei seguenti punti:

- Monitor debugger per le schede **GPC®** della **grifo®**, dotate di microprocessori Intel 51 (vedere elenco nei paragrafi successivi).
- Comunicazione su linea seriale di tipo asincrono, in **RS 232**.
- Protocollo fisico di comunicazione
 - 38400 Baud;**
 - 8 bit per carattere;**
 - Nessuna Parità;**
 - 1 Stop Bit.**

- **Linea seriale** di comunicazione **selezionabile** (tra **A** e **B**) su quelle schede che sono provviste di sue linee indipendenti.
- Utilizzabile con numerosi programmi di comunicazione per **PC**. In generale e' sufficiente un programma di emulazione terminale in grado di utilizzare il protocollo fisico descritto. Tra i questi si possono ricordare quelli appositamente preparati dalla **grifo®** (come il **GET51**) oppure sviluppati da terze parti (ad esempio **HYPERTERMINAL** di Windows).
- Numerosi **comandi** disponibili equivalenti ad altrettante funzionalita' (vedere elenco nei capitoli successivi).
- Possibilita' di scaricare programmi applicativi alla scheda, nel formato **HEX Intel** od **S Motorola**.
- **Veloce scaricamento** del programma (mediamente in **1 secondo** si trasferisce **1K Bytes** di codice equivalenti a circa **3K Bytes** di file **HEX**).
- Possibilita' di esecuzione del programma applicativo a **velocita' reale** od a **singola istruzione**.
- Gestione di **quattro punti d'arresto** diversi, che possono essere inseriti arbitrariamente dall'utente per controllare l'esecuzione del programma.
- Sovrapposizione di **area dati** ed **area codice** esterna del microcontrollore, in modo da consentire l'accesso in lettura e scrittura ad entrambi i tipi di memoria.
- Indirizza diversi tipi di memoria con le dimensioni massime che seguono:
 - fino a **55,75K Bytes** di **SRAM**;
 - fino a **31,75K Bytes** di **FLASH**.
- Possibilita' di **salvare** il programma utente nella memoria non volatile **FLASH**.
- Gestione modalita' **AUTORUN** in cui il programma applicativo salvato in **FLASH** parte **automaticamente** all'accensione o reset della scheda.
- Gestione modalita' **DEBUG** in cui all'accensione o reset della scheda parte sempre l'**FMO53**.
- Selezione modalita' di partenza (**AUTORUN** o **DEBUG**) tramite **DIP-SWITCH** o **Jumper** a bordo scheda.
- Utilizza **pochissime risorse** dell'hardware utilizzato, che sono comunque liberate in fase di esecuzione del programma applicativo utente.
- **Salvaguarda/ripristina** il contenuto di tutti i **registri** e della **RAM interna** quando prende/passa il controllo dal/al programma utente.
- **Rappresentazione ottimizzata** di informazioni su una videata del PC, che consente di mantenere una visione completa dello stato della scheda durante la fase di debug.
- Per i comandi che visualizzano grosse quantita' di dati sono previste delle azioni di **sospensione, ripresa** ed **abbandono** della rappresentazione.
- Completa gestione degli **interrupt** del microcontrollore che vengono ridirezionati in apposite aree di memoria codice; l'utente non deve modificare le procedure di risposta ma semplicemente riallocarle nelle suddette aree. Normalmente la riallocazione dell'inizio del codice e' sufficiente per **riallocare** anche tutti i **vettori d'interrupt**.
- Include procedure di **utilita'** per la scrittura di **aree dati** della **FLASH** direttamente da programma utente, fino ad un massimo di **7,75K Bytes**.
- **Protezioni e controlli ridondanti** sul contenuto della **FLASH** in modo da garantirne il contenuto (sia come codice che dati) in ogni condizione operativa.
- **Test** integrato della memoria **SRAM** sempre presente, richiamabile da apposito comando.
- Perfettamente **integrabile** negli ambienti di sviluppo per la famiglia I51, come il **BASCOM8051** (compilatore BASIC), **µC/51** (compilatore C), **SYS51PW** (compilatore

PASCAL), ecc.

- Il programma applicativo dell'utente da utilizzarsi in abbinamento all'**FMO53** richiede pochissimi accorgimenti:
 - definire l'**indirizzo di partenza** del **codice** e degli adiacenti vettori d'interrupt;
 - definire l'**indirizzo di partenza** dei **dati** usati dal programma.Tali settaggi con gli ambienti di sviluppo descritti sopra coincidono con due **direttive** da aggiungere nel **sorgente** o nel **progetto** dello stesso programma.
- Per gli ambienti di sviluppo proposti da **grifo**[®] l'utente trova una completa descrizione sia delle **configurazioni** per effettuare l'**integrazione** che delle **direttive** per preparare il programma applicativo.
- Ampia **documentazione** e notevole serie di **esempi** sia in formato **sorgente** che **eseguitibile**. Tali esempi coincidono con quelli delle schede di controllo, sono codificati in diversi linguaggi di programmazione e sono pronti per essere scaricati ed eseguiti.
- Il pacchetto e' composto da un **CD** che contiene software, manuale d'uso ed esempi e da una **EPROM** montata sulla scheda in uso.
- **Nessuna licenza** o costo aggiuntivo. L'utente e' libero di realizzare tutte le applicazioni che desidera.

Nel presente manuale é riportata una descrizione di tutte le caratteristiche dell'**FMO53** sufficienti per un suo uso. Si consiglia comunque anche la lettura del manuale tecnico della scheda usata in cui si trovano le informazioni sulla sua configurazione e modalit  d'uso; quest'ultime sono usate nei successivi paragrafi dando per scontato che l'utente le conosca.

MATERIALE NECESSARIO

Viene di seguito riportata una breve descrizione del materiale (hardware e software), necessario per operare con l'**FMO53**:

SCHEDA DI CONTROLLO

Coincide con una scheda di controllo appartenente al carteggio industriale **grifo®**, basata sui microprocessori della famiglia I51 e dotata di memoria esterna come: **GPC® R63; GPC® T63; GPC® R63D; GPC® T63D; GPC® 323; GPC® 323D; GPC® 324; GPC® 324D; GPC® 550; GPC® 552; GPC® 553; GPC® 554**; ecc.

La scheda di controllo, indipendentemente dalle richieste dell'applicazione da realizzare, deve essere dotata di:

- almeno 32K Byte di SRAM
- una linea seriale asincrona in RS 232
- una EPROM con le seguenti indicazioni:

FMO53
Ver. ??
GPC xxxx

dove: xxx = codice scheda
 ?? = versione programma

Quanto sopra riportato è da intendersi come struttura minima di lavoro, infatti lo stesso sistema può essere espanso aumentando quindi le sue potenzialità. La scelta della configurazione della scheda di controllo deve comunque avvenire in relazione alle specifiche esigenze dell'applicazione che deve essere sviluppata. Tra queste opzioni ricordiamo:

- FLASH EPROM Atmel:

29c256 da 32K Bytes (.32KF)
29c010 da 128K Bytes (.128KF)

 per il salvataggio del programma su scheda.
- SRAM:

da 32K Bytes (.32K)
da 128K Bytes (.128K)

 per sviluppare programmi molto lunghi o con grosse aree dati.

In caso di acquisto contemporaneo di scheda di controllo, **FMO53** ed eventuali opzioni, i dispositivi di memoria vengono forniti già montati sulla scheda.

PERSONAL COMPUTER

Il pacchetto **FMO53** necessita di un personal computer, che da ora in poi chiameremo **PC di sviluppo**, con le seguenti caratteristiche:

<i>Personal Computer:</i>	IBM compatibile (con CPU ≥ 486).
<i>Memoria RAM:</i>	Minimo 640K Bytes.
<i>Sistema operativo:</i>	MS-DOS o WINDOWS 3.11, 95, 98, ME, 2000, NT, XP

Monitor:	Colori.
Memorie di massa:	Lettoce CD-ROM Hard Disk con almeno 2M Byte liberi.
Seriale:	RS 232, secondo specifiche V24
Mouse:	Microsoft compatibile con relativo driver installato.

CAVO DI COMUNICAZIONE SERIALE

Per tutte le fasi previste dall'**FMO53** é necessario effettuare un collegamento seriale tra una delle linee seriali del PC di sviluppo ed una delle linee seriali della scheda di controllo. Tale collegamento necessita solo dei segnali di ricezione, trasmissione e massa (RxD, TxD e GND) e deve avvenire seguendo le normative V24 del C.C.I.T.T.

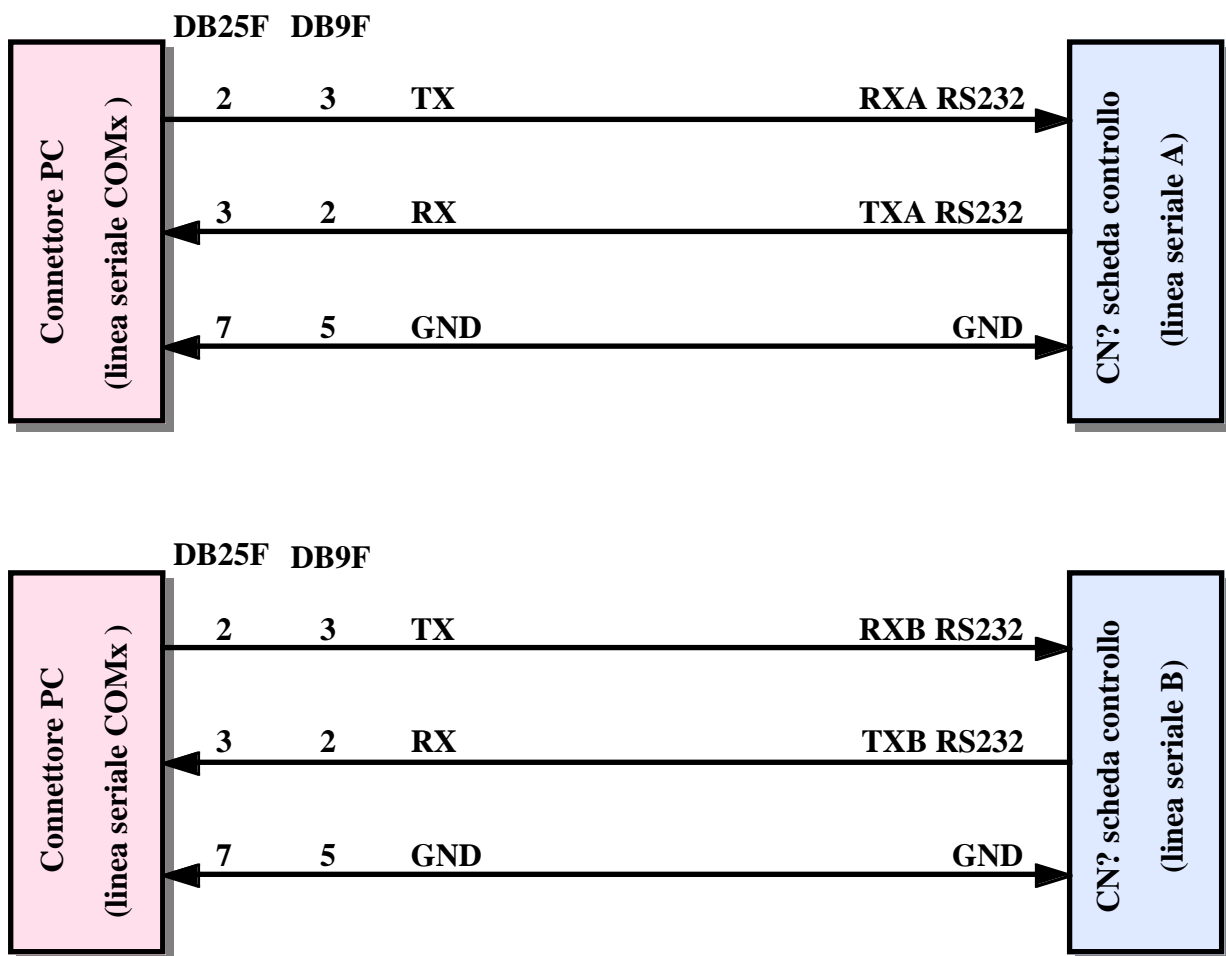


FIGURA 1: COLLEGAMENTO SERIALE TRA PC DI SVILUPPO E SCHEDA DI CONYROLLO

La figura 1 illustra il collegamento tra i connettori seriali normalizzati di un PC ed il connettore presente sulla scheda di controllo: di quest'ultima non sono indicati i numeri dei contatti del connettore in quanto può variare al variare della scheda.

Sul lato PC, qualora non sia disponibile alcuna linea di comunicazione seriale, si possono usare degli appositi convertitori che una volta aggiunti al PC in uso mettono a disposizione la linea seriale RS232 richiesta. Tra questi si possono citare i convertitori USB<->RS232, i convertitori ETHERNET<->RS232, le schede multi I/O con seriali RS232 aggiuntive, ecc. Naturalmente

l'FMO53 può usare questi dispositivi solo se correttamente installati sia dal punto di vista hardware che software, secondo le indicazioni della casa costruttrice.

Sul lato scheda di controllo, come indicato nella precedente figura, l'utente può selezionare quale linea utilizzare quando la scheda dispone di almeno due seriali. Questa caratteristica consente di sviluppare comodamente anche quelle applicazioni in cui una seriale è già usata (ad esempio per collegamenti in rete, a modem, a stampanti, altri dispositivi, ecc.) ed è abbondantemente descritta nel paragrafo SELEZIONE LINEA SERIALE.

Al fine di velocizzare la fase di collegamento ed eliminare la necessità di dover realizzare un cavo di comunicazione, la grifo® è in grado di offrire i cavi ed accessori di comunicazione seriale già pronti, per qualsiasi tipo di scheda e PC:

Scheda controllo	Seriale	CN?	Codici accessori	Codici cavi seriali
GPC® R63 GPC® T63 GPC® R63D GPC® T63D	A	CN3	MSI 01; AMP8.Cable	CCR.PLUG25F o CCR.PLUG9F
GPC® 323 GPC® 323D	A	CN3A	-	CCR.PLUG25F o CCR.PLUG9F
	B	CN3B	-	CCR.PLUG25F o CCR.PLUG9F
GPC® 324	A	CN3A	-	CCR.PLUG25F o CCR.PLUG9F
GPC® 324D	A	CN3A	-	CCR.PLUG25F o CCR.PLUG9F
	B	CN3B	-	CCR.PLUG25F o CCR.PLUG9F
GPC® 550	A	CN4A	-	CCR.PLUG25F o CCR.PLUG9F
	B	CN4B	-	CCR.PLUG25F o CCR.PLUG9F
GPC® 552	A	CN7	-	CCR.9+9R
	B	CN5	-	-
GPC® 553	A	CN3A	-	CCR.PLUG25F o CCR.PLUG9F
	B	CN3B	-	CCR.PLUG25F o CCR.PLUG9F
GPC® 554	A	CN3A	-	CCR.PLUG25F o CCR.PLUG9F
	B	CN3B	-	CCR.PLUG25F o CCR.PLUG9F

FIGURA 2: CONNETTORE ED ACCESSORI PER COLLEGAMENTO SERIALE

La tabella descrive per tutte le schede di controllo, i connettori su cui sono disponibili le linee seriali, gli eventuali accessori ed infine il cavo seriale che direttamente, o indirettamente tramite gli accessori, effettua la connessione descritta in figura 1. I numeri finali nel codice del cavo indicano se la vaschetta femmina per il PC è a 25 o 9 vie compatibilmente con gli standard in uso. L'utente può quindi decidere se realizzare i cavi di collegamento autonomamente oppure ordinarli direttamente alla grifo®.

SOFTWARE DI LAVORO

Assieme all'hardware descritto per operare con l'FMO53 sono necessari dei software di lavoro con cui sviluppare e mettere a punto il programma applicativo. Tale software è organizzato sotto forma di appositi pacchetti e può essere suddiviso in due gruppi principali, come di seguito descritto.

AMBIENTE DI SVILUPPO PROGRAMMA APPLICATIVO

Prima di essere trasferito alla scheda di controllo e su questa provato e salvato, il programma applicativo dell'utente deve essere generato. A questo scopo si possono convenientemente usare degli appositi ambienti di sviluppo che consentono di scrivere il programma comodamente sul PC di sviluppo e di trasformarlo nel codice macchina usato dall'**FMO53**.

In generale la scheda può sfruttare tutte le risorse software per il microcontrollore montato, ovvero i numerosi pacchetti ideati per la famiglia I51, sia ad alto che a basso livello. Tutti i pacchetti di sviluppo software forniti dalla **grifo**® sono sempre accompagnati da esempi, in formato sorgente ed eseguibile, pronti all'uso con **FMO53**. Tra questi ricordiamo:

HI TECH C 51: Cross compilatore per file sorgenti scritti in linguaggio C. E' un potente pacchetto software che tramite un comodo IDE permette di utilizzare un editor, un compilatore C (floating point), un assembler, un ottimizzatore, un linker e un remote debugger. Sono inoltre inclusi i source delle librerie.

SYS51CW: Cross compilatore per programmi scritti in C, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione: editor, compilatore C, assembler, ottimizzatore, linker, librerie ed un debugger simbolico remoto.

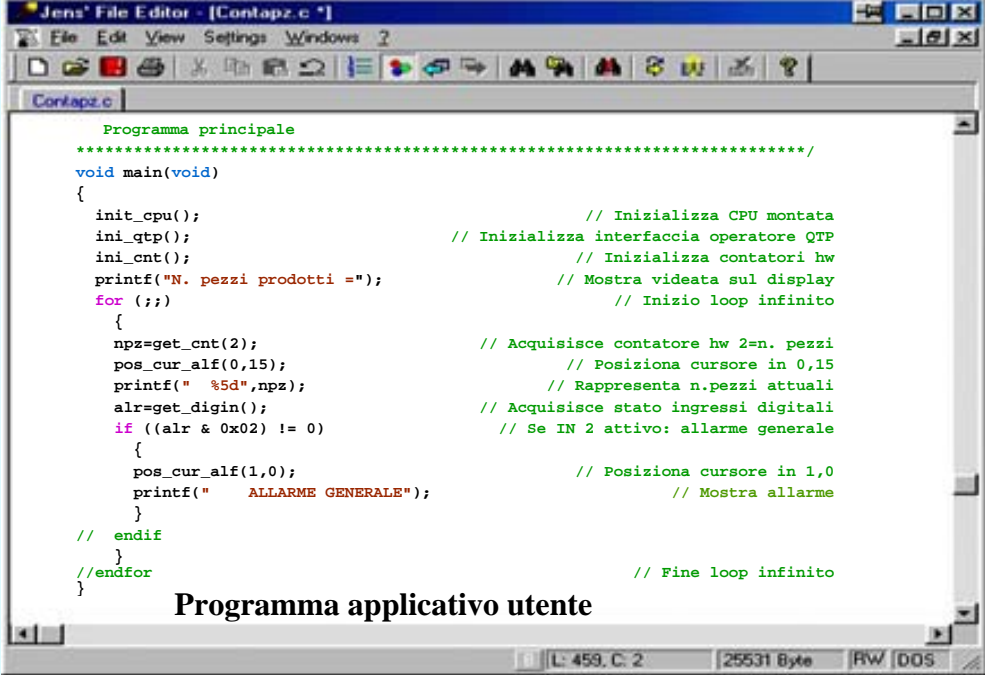
SYS51PW: Cross compilatore per programmi scritti in PASCAL, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione: editor, compilatore PASCAL, assembler, ottimizzatore, linker, librerie ed un debugger simbolico remoto.

DDS MICRO C 51: E' un comodo pacchetto software, a basso costo, che tramite un completo IDE permette di utilizzare un editor, un compilatore C (integer), un assembler, un linker e un remote debugger abbinato ad un monitor. Sono inclusi i sorgenti delle librerie ed una serie di utility.

BASCOM 8051: Cross compilatore a basso costo per files sorgenti scritti in BASIC, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore ed un simulatore molto potente per il debugger del sorgente. Comprende molti modelli di memoria, svariati tipi di dati e numerose istruzioni dedicate alle tipiche risorse hardware usate nell'automazione industriale.

µC/51: E' un comodo pacchetto software, a basso costo, che tramite un completo IDE permette di utilizzare un editor, un compilatore ANSI C, un assembler, un linker e un remote debugger configurabile da utente, a livello sorgente. Sono inclusi i sorgenti delle librerie fondamentali e del remote debugger, alcuni esempi di utilizzo e vari programmi di utility.

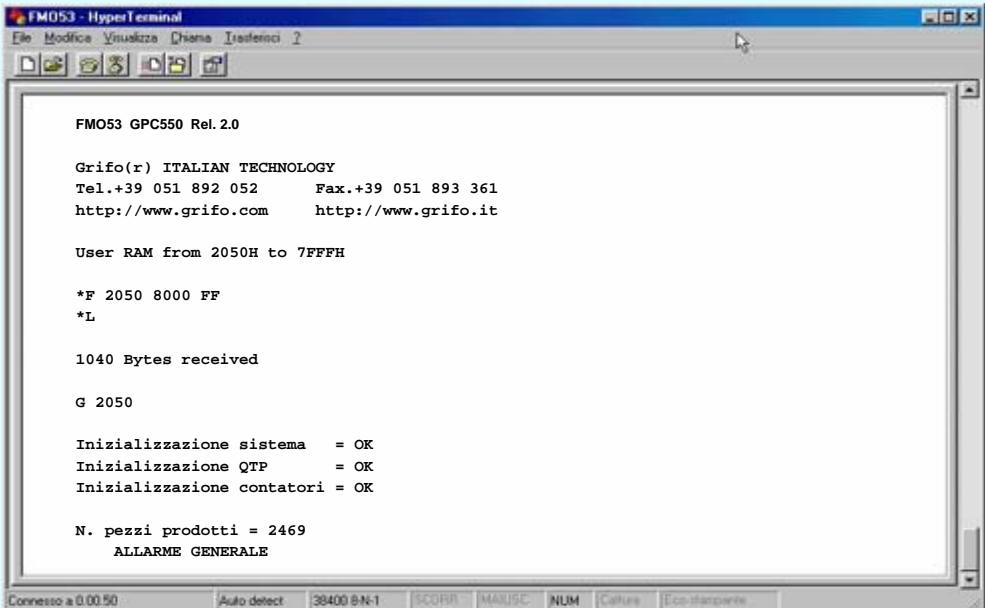
LADDER WORK: E' un semplice sistema per creare programmi di automazione con la conosciuta e diffusa logica a contatti. Include un editor grafico che consente di posizionare e collegare i componenti hardware della scheda (input, output, contatori, A/D, ecc) come su uno schema elettrico e di definirne le proprietà, un efficiente compilatore che converte lo schema in codice eseguibile ed utility per il download di tale codice verso la scheda. Il tutto integrato in un comodo IDE per Windows. Viene fornito sotto forma di CD che comprende esempi e manuale d'uso e relativa chiave di abilitazione.



Programma applicativo utente

Ambiente sviluppo

+



Emulazione terminale seriale

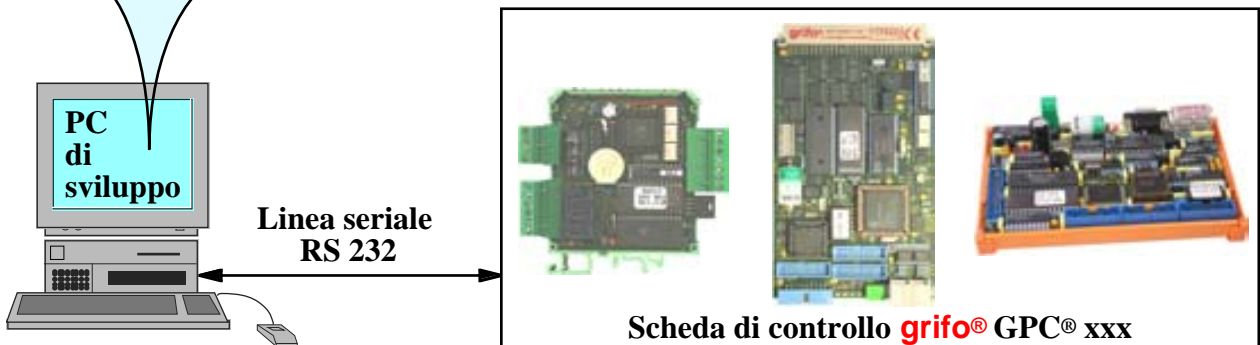


FIGURA 3: MODALITÀ DI LAVORO

PROGRAMMA EMULAZIONE TERMINALE SERIALE

Coincide con un generico programma di comunicazione in grado di gestire una classica emulazione terminale, con un protocollo fisico di comunicazione impostabile. Tale programma é la naturale controparte dell'**FMO53** e deve rappresentare sul monitor tutto quanto ricevuto dalla linea seriale, trasmettere sulla stessa seriale quanto premuto sulla tastiera ed infine trasmettere un file salvato sul PC di sviluppo. Con parole alternative il PC di sviluppo che esegue tale programma, si comporta come una console per la scheda di controllo, con cui l'utente può facilmente usare tutte le funzionalità dell'**FMO53**.

A questo scopo si ricorda il **GET51** sviluppato dalla **grifo®**, il famoso **HYPERTERMINAL** di Windows, od i diffusi programmi **CROSS TALK**, **PROCOMM**, **BITCOMM**, **TERMINAL**, ecc. realizzati da terze parti.

PROGRAMMATORE DI EPROM

Un programmatore di EPROM in grado di programmare file presenti sul PC di sviluppo é necessario per completare l'applicazione realizzata. Infatti il codice generato, una volta debuggato e provato in tutte le sue parti, deve essere salvato permanentemente su una EPROM da montare sulla scheda di controllo.

Da ricordare che il programmatore di EPROM é necessario solo se la scheda usata non è dotata di FLASH EPROM (opzione .32KF o .128KF). Infatti in quest'ultimo caso il salvataggio sulla stessa FLASH é autonomamente svolta dalla scheda di controllo, tramite il PC di sviluppo ed un apposito comando dell'**FMO53**.

DESCRIZIONE FMO53

In questo capitolo vengono descritte tutte le caratteristiche dell'**FMO53** relative sia al suo uso a bordo della scheda di controllo, che alla generazione del programma applicativo. Se l'utente desidera velocizzare il primo utilizzo del pacchetto, si consiglia di seguire i passi del capitolo COME INIZIARE che, quando é necessario, rimanda ai paragrafi di questo capitolo.

CONFIGURAZIONE SCHEDA

Per poter sfruttare tutte le potenzialità dell'**FMO53** la scheda di controllo usata deve essere opportunamente configurata. Tale configurazione si rivolge principalmente alla presenza ed al mappaggio delle memorie di bordo, su cui si basano tutte le operazioni di scaricamento, prova e salvataggio del programma applicativo.

In generale l'**FMO53** deve essere in grado di poter leggere e scrivere sia l'area codice che l'area dati indirizzata dal microcontrollore e, visto che l'architettura dell'8051 non consente la scrittura dell'area codice, la scheda deve sovrapporre le due aree. In altre parole l'**FMO53** può scaricare il codice, gestire i punti d'arresto, eseguire il programma applicativo, testare la memoria, ecc. solo se la SRAM della scheda é indirizzata sia come dati che come codice (segnali /PSEN e /RD del microcontrollore sommati).

In dettaglio questa condizione operativa la si ottiene in modo diverso a seconda della scheda di controllo usata, e fisicamente corrisponde al montaggio di appropriati componenti di memoria ed alla configurazione di appositi jumpers e/o dip switch. Come indicato nel paragrafo SCHEDA DI CONTROLLO del capitolo MATERIALE NECESSARIO se l'**FMO53** viene ordinato assieme alla scheda, l'utente riceve l'hardware già configurato e pronto all'uso. Per ottenere tale condizione si deve specificare nell'ordine il codice **.FMO53**, come estensione del nome della scheda, eventualmente seguito dai codici delle opzioni di memoria. Ad esempio per ricevere una **GPC® 324D** con l'**FMO53** e 32K di FLASH pronti all'uso si deve ordinare il codice:

GPC® 324D.FMO53.32KF

Viceversa in caso di ordine separato, od effettuato in un secondo tempo, le tabelle di figura 4 e 5 illustrano come installare le memorie e configurare la scheda autonomamente, in modo da usare correttamente l'**FMO53**. In queste tabelle sono riportate numerose informazioni, brevemente descritte nella seguente legenda:

- | | |
|------------------------|---|
| - Scheda | indica per quale scheda di controllo valgono le istruzioni. |
| - Mappaggio Memorie | indica il nome del mappaggio memorie, così come riportato del manuale tecnico della scheda. |
| - Selezione Mappaggio | indica quali jumper e/o dip switch muovere per ottenere il mappaggio |
| - AUTORUN | indica l'impostazione per far partire automaticamente all'avvio il programma utente, salvato nella FLASH. |
| - EPROM FMO53 | indica in quale zoccolo inserire la EPROM dell' FMO53 . |
| - SRAM base | indica in quale zoccolo inserire la SRAM di base (se assente). |
| - Zoccolo multimemoria | indica in quale zoccolo inserire la memoria opzionale ed aggiuntiva, gestita dall' FMO53 . |
| - Jumpers Memorie | riporta la configurazione dei jumpers relativi alle memorie descritte. |

Per identificare la posizione degli zoccoli e dei jumpers descritti in queste tabelle, si prega di consultare le apposite figure del manuale tecnico della scheda.

Scheda	Mappaggio Memorie	Selezione Mappaggio	AUTO RUN	EPROM FMO53	SRAM base	Zoccolo multim.	Jumpers Memorie
GPC® R63 GPC® T63 GPC® R63D GPC® T63D Rel 101197	MODO 3	J2 chiuso J3 in 2-3	J1 chiuso	IC 7	IC12 saldata	SRAM (.32K) IC 10	J4 in 2-3 J5 in 2-3
						FLASH (.32KF) IC 10	J4 in 1-2 pin 3 di J4 con pin 2 di J5
GPC® R63 GPC® R63D Rel≥110901	MODO 3	J2 chiuso J3 in 2-3	J1 chiuso	IC 7	IC12 saldata	SRAM (.32K) IC 10	J5 in 2-3 e 4-5
						FLASH (.32KF) IC 10	J5 in 1-2 e 3-4
GPC® 323 GPC® 323D Rel 110197	MODO 1	DIP 5 ON DIP 6 ON DIP 7 OFF	DIP 8 ON	IC 5	IC 4	SRAM (.32K) IC 3	J2 in 2-3 J3 in 2-3 J4 e J5 in 1-2
						FLASH (.32KF) IC 3	J2 in 1-2 pin 3 di J2 con pin 2 di J3 J4 e J5 in 1-2
GPC® 323 GPC® 323D Rel≥250601	MODO 3	DIP 5 ON DIP 6 ON DIP 7 OFF	DIP 8 ON	IC 5	IC 4	SRAM (.32K) IC 3	J2 in 2-3 e 4-5
						FLASH (.32KF) IC 3	J2 in 1-2 e 3-4
GPC® 324 GPC® 324D Rel 100997	MODO 3	J2 chiuso J3 in 2-3	J1 chiuso	IC 8	IC7 saldata	SRAM (.32K) IC 5	J4 in 2-3 J5 in 2-3 J6 in 1-2
						FLASH (.32KF) IC 5	J4 in 1-2 pin 3 di J4 con pin 2 di J5 J6 in 1-2
GPC® 324 GPC® 324D Rel≥110400	MODO 3	J2 chiuso J3 in 2-3	J1 chiuso	IC 8	IC7 saldata	SRAM (.32K) IC 5	J4 in 2-3 e 4-5
						FLASH (.32KF) IC 5	J4 in 1-2 e 3-4

FIGURA 4: CONFIGURAZIONE SCHEDE (1 DI 2)

Scheda	Mappaggio Memorie	Selezione Mappaggio	AUTO RUN	EPROM FMO53	SRAM base	Zoccolo multim.	Jumpers Memorie
GPC® 550 Rel 200702	MODO 3	J4 in 2-3 J5 chiuso	DIP 8 ON	U1	U2 saldata	SRAM (.32K) U3	J1 in 2-3 e 4-5
						FLASH (.32KF) U3	J1 in 1-2 e 3-4
GPC® 550 Rel≥120605	MODO 3	J1 chiuso J2 chiuso J12 aperto	DIP 8 ON	IC19	IC7 saldata	SRAM (.32K) IC 18	J3 in 1-2 J7 in 2-4 e 5-6
						SRAM (.128K) IC 18	J3 in 1-2 J7 in 2-4;5-6;7-8
						FLASH (.32KF) IC 18	J3 in 1-2 J7 in 1-2;4-5;6-8
						FLASH (.128KF) IC 18	J3 in 1-2 J7 in 2-3;4-5;6-8
GPC® 552 Rel≥180796	MODO 1	J1 in 3 J2 in 3 J14 (1,2,3) in ASM	DIP 8 ON	IC 15	IC 13	SRAM (.32K) IC 12	J3 e J4 in 1-2 J17 in 2-3 J18 in 2-3
						FLASH (.32KF) IC 12	J3 e J4 in 1-2 pin 2 di J17 con pin 3 di J18 J18 in 1-2
GPC® 553 Rel≥070198	MODO 3	DIP 5 ON DIP 6 ON J7 aperto	DIP 8 ON	IC 1	IC 2	SRAM (.32K) IC 3	J2 in 2-3 J3 in 2-3 J4 e J5 in 1-2
						FLASH (.32KF) IC 3	J2 in 1-2 pin 2 di J3 con pin 3 di J2 J4 e J5 in 1-2
GPC® 554 Rel≥100997	MODO 3	J1 chiuso J6 in 2-3	J2 chiuso	IC 4	IC8 saldata	SRAM (.32K) IC 6	J3 in 2-3 J4 in 2-3 J5 in 1-2
						FLASH (.32KF) IC 6	J3 in 1-2 pin 3 di J3 con pin 2 di J4 J5 in 1-2

FIGURA 5: CONFIGURAZIONE SCHEDE (2 DI 2)



Una volta configurata completamente la scheda, come da precedenti indicazioni, l'FMO53 gestisce le risorse di memoria disponibili in due diverse modalità, come illustrato nelle seguenti figure.

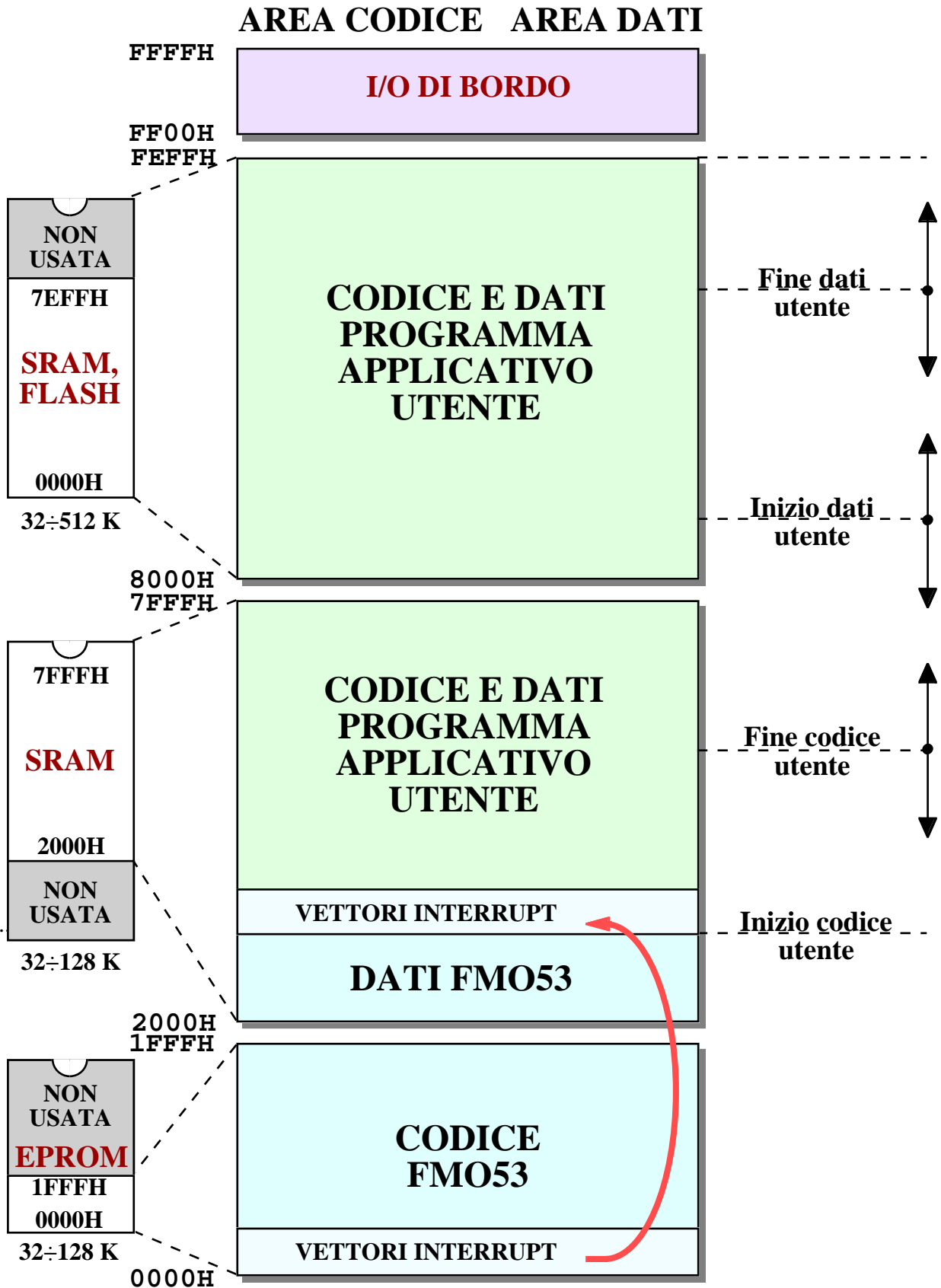


FIGURA 6: ORGANIZZAZIONE MEMORIE IN MODALITÀ DEBUG

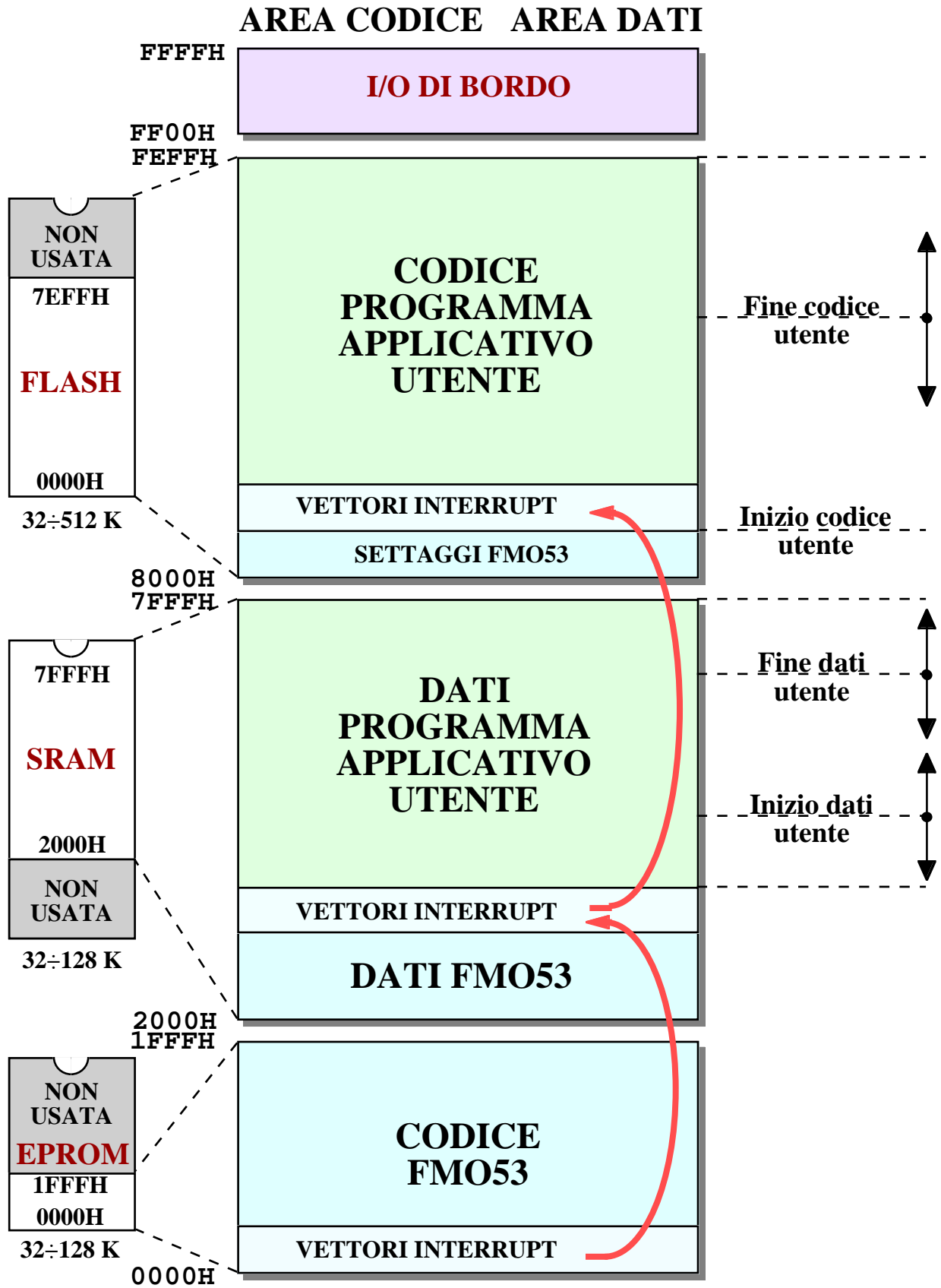


FIGURA 7: ORGANIZZAZIONE MEMORIE IN MODALITÀ AUTORUN

Le due modalità DEBUG ed AUTORUN sono le condizioni operative che l'**FMO53** seleziona a seguito di ogni partenza, come descritto nel successivo paragrafo SELEZIONE MODO OPERATIVO.

Le figure 6 e 7 riportano tutte le informazioni necessarie per sviluppare il proprio programma applicativo in abbinamento all'**FMO53**, infatti descrivono gli indirizzi di inizio e fine del codice e dei dati a disposizione dell'utente, le ridirezioni degli interrupts e le allocazioni di queste aree sui dispositivi fisici di memoria. Sulle stesse figure si possono fare le seguenti precisazioni:

- a) Molte indicazioni sono generiche in quanto variano a seconda delle condizioni operative; per avere i valori reali degli indirizzi si consiglia di esaminare il paragrafo CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE e le figure 9, 10.
- b) Le doppie frecce verticali, che affiancano alcuni indirizzi di inizio e fine, indicano che lo stesso indirizzo può variare dall'adiacente estremo inferiore fino al successivo estremo superiore.
- c) La memoria EPROM indirizzata da 0000÷1FFFH corrisponde alla colonna *EPROM FMO53* delle tabelle di figura 4 e 5.
- d) La memoria SRAM indirizzata da 2000÷7FFFH corrisponde alla colonna *SRAM base* delle tabelle di figura 4 e 5.
- e) La memoria SRAM o FLASH indirizzata da 8000÷FEFFH corrisponde alla colonna *Zoccolo multimemoria* delle tabelle di figura 4 e 5.
- f) La memoria FLASH può essere scritta solo tramite un apposito comando o procedura dell'**FMO53**, ma non direttamente dal microcontrollore, né dal programma utente.
- g) Il programma applicativo utente può essere eseguito e debuggato solo se è salvato su memoria SRAM.

RISORSE UTILIZZATE

Per svolgere il suo lavoro l'**FMO53** usa pochissime risorse della scheda che lo esegue, come elencato di seguito:

- **6 Bytes** di **RAM interna** (08H÷0DH) per sue variabili di lavoro;
- **208 Bytes** di **SRAM esterna** per salvataggio stato microcontrollore, punti d'arresto, ecc.;
- **8K Bytes** di **EPROM** per il suo codice;
- **20 Bytes** di **stack**;
- **80 Bytes** di **FLASH** per i settaggi di AUTORUN e controlli validità del programma utente;
- una **linea seriale** in RS 232 per la comunicazione nei confronti del PC di sviluppo;
- **Timer1** per la comunicazione seriale e l'esecuzione passo passo;
- **interrupt Timer1** per l'esecuzione passo passo;
- **interrupt Timer1** ed **/INT0** per la comunicazione con seriale B software;
- un **selettore** hardware per la selezione della modalita' di partenza;
- una **configurazione** per la selezione della seriale usata.

Durante la programmazione della FLASH con il programma utente vengono usate ulteriori risorse di RAM interna, SRAM esterna e l'interrupts della linea seriale hardware usata. Visto che tale programmazione coincide con la fase conclusiva del processo di sviluppo del programma utente, a cui normalmente segue un reset od una riaccensione della scheda, all'utente può disinteressarsi dell'uso di queste risorse.

Da ricordare che tutte le risorse vengono completamente liberate quando si esegue il programma salvato in FLASH; l'utente può quindi lavorare come se il suo programma fosse da solo.

COMUNICAZIONE SERIALE

L'interazione tra **FMO53** ed utente é completamente basata su una comunicazione seriale asincrona che avviene tra la scheda di controllo ed il PC di sviluppo. Nel paragrafo CAVO DI COMUNICAZIONE SERIALE é descritto il collegamento elettrico tra i due sistemi mentre in questo si riportano le rimanenti informazioni, relative alla comunicazione seriale

Il protocollo fisico usato dall'**FMO53**, che deve essere naturalmente impostato anche sul PC, é il seguente:

Baud Rate	=	38400
Bit per carattere	=	8
Parità	=	Nessuna
Stop Bit	=	1

Sul lato scheda di controllo, l'utente può selezionare quale linea utilizzare quando la scheda dispone di almeno due seriali. Questa caratteristica consente di sviluppare comodamente anche quelle applicazioni in cui una seriale é già usata (ad esempio per collegamenti in rete, a modem, a stampanti, altri dispositivi, ecc.). Inoltre permette di usare il PC di sviluppo collegato in RS 232 anche quando la rimanente linea della scheda é configurata in RS 422, RS 485 o Current loop, senza dover interporre costosi convertitori od effettuare scomode variazioni di configurazione.

La scelta della linea seriale da utilizzare é effettuata tramite un'apposita condizione hardware sulla scheda di controllo, che viene verificata in corrispondenza della partenza dell'**FMO53**, ovvero dopo un reset od un power on. Tale condizione varia al variare della scheda, come sotto descritto:

Scheda controllo	Seriale	CN?	Condizione per selezione della seriale usata	Configurazione per seriale scelta
GPC® R63 GPC® T63 GPC® R63D GPC® T63D	A	CN3	-	-
GPC® 323	A	CN3A	DSW1.4=OFF	J22,J23=Connessi; J24 in 1-2
	B	CN3B	DSW1.4=ON	J18 in 2-3
GPC® 323D	A	CN3A	DSW1.4=OFF	J22,J23=Connessi; J24 in 1-2
	B	CN3B	DSW1.4=ON	J18 in 1-2
GPC® 324	A	CN3A	-	J9 in 1-2; J10,J11=Connessi
GPC® 324D	A	CN3A	Pin 1,2 di CN5=Non connessi	J9 in 1-2; J10,J11=Connessi
	B	CN3B	Pin 1,2 di CN5=Connessi	-
GPC® 550	A	CN4A	DSW1.4=OFF	J11,J13=Connessi; J17 in 2-3
	B	CN4B	DSW1.4=ON	J5 in 2-4
GPC® 552	A	CN7	DSW1.4=OFF	J19,J20,J21=Connessi
	B	CN5	DSW1.4=ON	J10 in 1-2; J16 in 1-2
GPC® 553	A	CN3A	DSW1.4=OFF	JS7,JS8=Connessi; JS9 in 1-2
	B	CN3B	DSW1.4=ON	J12 in 2-3; J13 in 2-3
GPC® 554	A	CN3A	J7=Non connesso	-
	B	CN3B	J7=Connesso	JS12 in 2-3; JS13 in 2-3

FIGURA 8: SELEZIONE LINEA SERIALE USATA

Sulle schede in cui non é riportata la condizione, l'**FMO53** comunica sempre e solo con la linea seriale A. La tabella riporta anche le configurazioni che rendono funzionante la linea seriale scelta: l'utente deve effettuare queste configurazioni aiutandosi con il manuale tecnico della scheda.

CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE

In tutti i programmi applicativi si possono sempre identificare due spazi di memoria distinti che vengono definiti:

- area codice** -> destinata al codice delle istruzioni del programma, alle costanti, messaggi, librerie, run time, ecc. Il microcontrollore gestisce tale area solo in lettura;
- area dati** -> destinata a tutti i dati del programma ovvero alle variabili semplici e strutturate, stringhe, buffers, ecc. Il microcontrollore gestisce tale area sia in lettura che scrittura.

Partendo da questa premessa e dalle figure 6 e 7, in questo paragrafo vengono descritte le configurazioni che l'utente deve effettuare sul proprio programma applicativo, per renderlo compatibile con l'**FMO53**.

In generale sono sempre necessarie le seguenti azioni:

- Definire l'indirizzo di **Inizio codice utente** e degli adiacenti vettori d'interrupt.
- Verificare che la dimensione dell'area codice non sia eccessiva, ovvero che l'indirizzo di **Fine codice utente** assuma un valore ammesso.
- Definire l'indirizzo di **Inizio dati utente**.
- Verificare che la dimensione dell'area dati non sia eccessiva, ovvero che l'indirizzo di **Fine dati utente** assuma un valore ammesso.

Mentre sono consigliati i seguenti accorgimenti:

- Se il programma applicativo prevede una console seriale sulla stessa linea collegata al PC di sviluppo, é suggerito impostare nel programma lo stesso protocollo fisico dell'**FMO53** (38400,8,No,1).
- Durante la fase di sviluppo e messa a punto del programma, se si utilizzano i comandi di debug (punti di arresto, esecuzione a singolo passo, esame della memoria, ecc), é suggerito non usare le risorse utilizzate dall'**FMO53**.

I valori degli indirizzi di inizio e fine riportati nella precedente descrizione assumono diversi valori a seconda del modo operativo selezionato, della linea di comunicazione prescelta e del tipo di memoria installato sullo zoccolo multimemoria, come indicato nelle due successive tabelle:

Seriale usata	Zoccolo multim.	Inizio codice utente	Fine codice utente	Inizio dati utente	Fine dati utente
A	SRAM (.32K)	2050H	< Inizio dati utente e ≤ FEFFH	Fine codice utente ÷ FEFFH	≤ FEFFH
	FLASH (.32KF)	2050H	< Inizio dati utente e ≤ 7FFFH	Fine codice utente ÷ 7FFFH	≤ 7FFFH
B	SRAM (.32K)	2100H	< Inizio dati utente e ≤ FEFFH	Fine codice utente ÷ FEFFH	≤ FEFFH
	FLASH (.32KF)	2100H	< Inizio dati utente e ≤ 7FFFH	Fine codice utente ÷ 7FFFH	≤ 7FFFH

FIGURA 9: INDIRIZZI AREE DI MEMORIA IN MODALITÀ DEBUG

Seriale usata	Zoccolo multim.	Inizio codice utente	Fine codice utente	Inizio dati utente	Fine dati utente
A	FLASH (.32KF)	8050H	≤ FEFFH	20C6H ÷ 7FFFH	≤ 7FFFH
B	FLASH (.32KF)	8050H	≤ FEFFH	2176H ÷ 7FFFH	≤ 7FFFH

FIGURA 10: INDIRIZZI AREE DI MEMORIA IN MODALITÀ AUTORUN

Come si può notare da entrambe le tabelle, numerosi indirizzi non hanno un valore definito bensì un range di variazione. In questi casi l'utente può attribuirgli il valore che preferisce in base alle sue esigenze ed a possibili facilitazioni d'uso. Ad esempio con un programma applicativo che ha una lunghezza codice di 15K Bytes ed una lunghezza dati di 2K Bytes, è conveniente settare l'**Inizio dati utente** = 7000H in modo da mantenerlo invariato sia in modalità DEBUG che AUTORUN ed in modo da lasciare spazio di crescita sia per il codice che per i dati.

Sempre esaminando le tabelle di figura 9 e 10 si ricavano le dimensioni massime dell'area codice e dati del programma applicativo, in abbinamento all'**FMO53**:

Area	Dimensione massima in modalità DEBUG	Dimensione massima in modalità AUTORUN
Codice utente	55,6K Bytes - Dati utente (2050H ÷ FEFFH)	31,6K Bytes (8050H ÷ FEFFH)
Dati utente	55,6K Bytes - Codice utente (2050H ÷ FEFFH)	23,8K Bytes (20C6H ÷ 7FFFH)

FIGURA 11: DIMENSIONI MASSIME AREE DI MEMORIA

Qualora le dimensioni massime sopra riportate risultino insufficienti per l'applicazione da sviluppare, si ricorda che tutte le schede di controllo dispongono di altri mappaggi che mettono a disposizione quantità superiori di memoria. In questi casi particolari l'utente può continuare ad usare l'**FMO53** in modalità DEBUG per testare e mettere a punto il suo programma (eventualmente frazionandolo in modo da rientrare nei limiti massimi) ed al termine salvarlo su EPROM o FLASH EPROM con un programmatore esterno, rinunciando alla modalità AUTORUN. Per dettagli relativi a queste possibilità si consiglia di contattare direttamente la **grifo®**.

I settaggi degli indirizzi di inizio e fine dell'area codice e dati normalmente coincidono con delle **direttive** da aggiungere nel **sorgente** o nel **progetto** del programma applicativo. Naturalmente tali direttive si differenziano a seconda dell'ambiente di sviluppo usato e spesso vengono identificate da nomi che si riferiscono all'inizio della ROM e della RAM. Per gli ambienti di sviluppo proposti da **grifo®** l'utente può trovare una completa descrizione di tutti i settaggi che configurano il programma applicativo utente, nel successivo capitolo INTEGRAZIONE CON AMBIENTE DI SVILUPPO. Inoltre gli stessi settaggi sono presenti in molti degli esempi, forniti sia in formato sorgente che eseguibile, in modo da poterli provare immediatamente e quindi riutilizzarli per realizzare il proprio programma.

GESTIONE FLASH EPROM

Una delle caratteristiche fondamentali dell'**FMO53** é la sua possibilità di gestire l'eventuale FLASH EPROM presente sulla scheda di controllo. Nei precedenti paragrafi sono state descritte le sigle dei componenti, i codici d'ordine, il mappaggio e gli indirizzi di allocazione, mentre in questo vengono descritte le modalità di gestione di questa memoria.

Innanzitutto si ricorda che la FLASH è suddivisa in 3 parti:

- 8000H÷804FH** -> spazio riservato per settaggi di AUTORUN e controlli validità
- 8050H÷FEFFH** -> spazio per codice utente
- E000H÷FEFFH** -> spazio per dati su FLASH

di cui le prime due sono gestite direttamente con i comandi di salvataggio del programma applicativo mentre l'ultima é gestita da un'apposita procedura di utilità generale. Su tutte le parti sono comunque previsti delle **protezioni** e dei **controlli ridondanti** in modo da garantirne il contenuto in ogni condizione operativa.

CODICE SU FLASH

Per salvare il programma applicativo dell'utente in FLASH EPROM si devono eseguire i seguenti passi:

- C1) Generare il programma con le configurazioni per la modalità di AUTORUN (vedi paragrafo precedente), ovvero assegnare il valore 8050H all'indirizzo di **Inizio codice utente**.
- C2) Verificare che la configurazione hardware della scheda sia corretta (vedi figure 4 e 5).
- C3) Abilitare i comandi dedicati alla FLASH con l'apposito comando **H**.
- C4) Cancellare il precedente contenuto della FLASH con l'apposito comando **X**.
- C5) Scaricare il file HEX Intel od S Motorola ottenuto al punto C1 alla scheda di controllo, tramite l'apposito comando **L**.
- C6) Verificare che l'**FMO53** presenti un messaggio di programmazione corretta e relativa attivazione del settaggio di AUTORUN

Per maggiori informazioni si consiglia di esaminare la descrizione dei comandi, nell'omonimo capitolo.

Qualora il programma applicativo da salvare in FLASH utilizzi anche lo spazio per dati, sempre su FLASH, si deve verificare che il primo non si estenda sul secondo; infatti le due aree occupano uno spazio d'indirizzamento comune.

DATI SU FLASH

L'**FMO53** include una procedura direttamente richiamabile dal programma applicativo utente, che consente di salvare valori nello spazio per dati su FLASH. Gli utilizzi tipici di questa caratteristica sono il salvataggio permanente di parametri, settaggi di lavoro, dati di produzione, storici di funzionamento, ecc., che possono poi essere prelevati ed usati dallo stesso programma applicativo o da altri programmi.

Lo spazio per dati su FLASH é limitato a **7,75K Bytes**, é organizzato in 124 pagine di 64 bytes ed é disponibile concordemente alle dimensioni del codice salvato sulla stessa FLASH. Dagli indirizzi riportati ad inizio pagina si ricava infatti che lo spazio per dati su FLASH può essere utilizzato anche per il codice utente.

Per salvare dati su FLASH il programma applicativo dell'utente deve eseguire i seguenti passi:

D1) Verificare che la configurazione hardware della scheda sia corretta (vedi figure 4 e 5).

D2) Impostare i parametri d'ingresso della procedura di scrittura dati su FLASH, settando delle apposite locazioni in SRAM esterna, organizzate con la seguente corrispondenza:

2001H÷2041H	->	64 bytes che verranno memorizzati su FLASH
2041H	->	numero pagina da scrivere (00H÷7BH)
2042H	->	check sum di tutti i parametri d'ingresso, ottenuto dalla somma di tutti i valori impostati nelle locazioni da 2001H÷2041H (64 bytes e numero pagina) senza riporto, <u>incrementata di 1</u> .

Ad esempio, per azzerare tutte le 64 locazioni nella pagina 8 dello spazio dati su FLASH, si devono impostare i partametri con i valori sotto riportati:

2001H÷2041H	->	bytes	=	00H
2041H	->	numero pagina	=	08H
2042H	->	check sum	=	09H

D3) Eseguire la procedura effettuando una chiamata (**LCALL**) all'indirizzo **1A00H** e ricordando che vengono utilizzati fino a 15 bytes di stack.

D4) Esaminare l'esito della chiamata alla procedura di scrittura, verificando il contenuto della locazione 2041H, secondo la seguente corrispondenza:

80H	->	scrittura eseguita correttamente
AAH	->	numero pagina non valido
BBH	->	checksum parametri d'ingresso errato
CCH	->	dimensione della FLASH non valida
EEH	->	malfunzionamento della FLASH

Note:

- Il comando di cancellazione FLASH cancella l'intero componente, quindi a seguito della sua esecuzione, tutti i dati salvati su FLASH sono definitivamente persi.
- La procedura di scrittura dati su FLASH può essere utilizzata solo con un dispositivo da 32K Bytes (opzione .32KF), mentre con tutte le rimanenti dimensioni la procedura restituisce il codice di errore CCH.
- Ricordare che la memoria FLASH ha un numero finito di scritture: per questo i dati da salvare devono essere opportunamente scelti in modo che la loro scrittura, effettuata con la procedura descritta, non avvenga frequentemente.

SELEZIONE MODO OPERATIVO

L'**FMO53** prevede due modalità operative appositamente progettate per coprire tutte le possibili esigenze dell'utente. La verifica del modo operativo da usare viene effettuata solo alla partenza (ovvero all'accensione o dopo un reset della scheda), ed al termine di questa verifica l'**FMO53** entra nella modalità individuata, senza possibilità di variarla da parte dell'utente, ad esempio con un comando.

Le caratteristiche delle modalità operative sono state più volte descritte nei precedenti paragrafi e figure, quindi in questo paragrafo si descrivono le condizioni che le selezionano:

DEBUG Questa modalità viene scelta quando nella FLASH EPROM non è stato salvato un programma applicativo utente, oppure quando l'apposito selettore hardware, è impostato nell'omonima condizione DEBUG.

In questo modo operativo viene sempre eseguito l'**FMO53** e l'utente ha la possibilità di usare tutte le sue funzioni, consentendogli di sviluppare un nuovo programma, aggiornarne uno già installato, vedere gli esiti del funzionamento, correggere eventuali errori, ecc.

AUTORUN Questa modalità viene scelta quando nella FLASH EPROM è stato salvato un programma applicativo utente e l'apposito selettore hardware è impostato nell'omonima condizione RUN.

In questo modo operativo viene eseguito automaticamente il programma applicativo precedentemente salvato in FLASH, consentendo all'utente di eliminare il PC di sviluppo ed installare la scheda di controllo sul sistema da realizzare.

Il selettore hardware descritto sopra varia al variare della scheda usata, come descritto nella colonna AUTORUN, delle tabelle di figura 4 e 5. Su tutte le schede è stato sempre scelto un selettore di facile accesso ed utilizzo (Dip switch o Jumper), in modo da consentire comodamente all'utente di variare la modalità operativa.

INTERRUPTS

L'**FMO53** ridireziona tutti i vettori d'interrupt del microcontrollore a partire dall'indirizzo **Inizio codice utente**, che varia al variare della modalità operativa selezionata, come descritto nelle figure 6, 7, 9, 10. La ridirezione si rende necessaria in quanto i vettori originali del microcontrollore sono allocati a partire da 0000H dove risiede la EPROM dell'**FMO53** e la scelta di ridirezionarli a questo indirizzo, rispecchia l'organizzazione standard dell'8051, in cui i vettori d'interrupt seguono il vettore di reset da cui parte il codice utente.

A titolo di esempio il vettore per l'interrupt del Timer0, originariamente allocato all'indirizzo 000BH, viene ridirezionato a 205BH in caso di modalità DEBUG su seriale A, a 210BH in caso di modalità DEBUG su seriale B ed a 805BH in caso di modalità AUTORUN.

Quando il programma applicativo utente usa gli interrupt è sufficiente allocare le entrate alle procedure di risposta ai vettori ridirezionati, mentre le procedure di risposta non subiscono alcuna variazione. In caso di tempistiche stringenti, si deve ricordare che la ridirezione effettuata provoca un ritardo nell'esecuzione della procedura di risposta, quantificabile in 2,2 μ sec.

Normalmente utilizzando un ambiente di sviluppo ad alto livello (compilatori C, BASIC, ecc.) l'allocazione dei vettori d'interrupt viene fatta automaticamente dal compilatore e il programmatore non deve preoccuparsene.

Si deve inoltre ricordare che in modalità DEBUG alcuni interrupts sono usati dall'**FMO53**, come descritto nel paragrafo RISORSE UTILIZZATE.

INTEGRAZIONE CON AMBIENTE DI SVILUPPO

Al fine di semplificare e velocizzare l'uso dell'**FMO53**, normalmente si integra lo stesso programma con l'ambiente di sviluppo usato, in modo da ottenere un'unica struttura di lavoro. L'integrazione può essere facilmente realizzata con la maggioranza degli ambienti di sviluppo in commercio e qualora questi non lo consentano è sempre possibile usare le due strutture separatamente, con una piccola complicazione nell'uso, che comunque non pregiudica i benefici complessivi di utilizzo.

In generale l'**FMO53** si integra con tutti i programmi per PC in grado di gestire una emulazione terminale seriale, con il protocollo fisico indicato nel paragrafo COMUNICAZIONE SERIALE, e di scaricare un file salvato sullo stesso PC sulla stessa linea seriale. Nei successivi paragrafi viene descritta l'integrazione con alcuni di questi programmi, presentando il settaggio della comunicazione seriale e le configurazioni per la generazione del programma applicativo per gli ambienti di sviluppo provvisti di compilatore ed il solo settaggio della comunicazione per i programmi di semplice emulazione terminale.

UTILIZZO CON BASCOM 8051

BASCOM 8051 è un cross compilatore BASIC, economico, efficiente, facile da usare, provvisto di un completo IDE (Integrated Development Environment = ambiente di sviluppo integrato), che gira sotto qualunque versione di Windows e genera codice eseguibile per microcontrollori della famiglia 'I51. Per ulteriori informazioni sulle caratteristiche generali di questo prodotto si consiglia di esaminare il relativo foglio tecnico, il manuale d'uso oppure il completo aiuto in linea. Le informazioni relative all'integrazione con **FMO53** sono invece riportate nei seguenti paragrafi.

CONFIGURAZIONI PER COMUNICAZIONE SERIALE

Una delle numerose funzionalità dell'IDE **BASCOM 8051** coincide con un emulatore terminale seriale che gira in una finestra Windows ed è completamente configurabile ed adattabile alle specifiche di funzionamento dell'**FMO53**. In aggiunta l'emulatore terminale è in grado di scaricare automaticamente alla scheda, il codice del programma applicativo, ottenuto dall'ultima compilazione. In questo modo l'**FMO53** può ricevere dal **BASCOM 8051** il comando di scaricamento di un file HEX Intel, il file stesso (ovvero l'ultimo programma compilato) ed i caratteri di fine scaricamento, semplicemente attivando la voce di un apposito menù.

Per ottenere questo risultato bisogna usare l'IDE del **BASCOM 8051** nel seguente modo:

- 1) Tramite il menu *Options / Communications*, selezionare la porta seriale COMx usata sul PC, configurare i parametri del protocollo fisico di comunicazione dell'**FMO53**, disabilitare gli handshake hardware, come illustrato in figura 12 e premere *OK*.

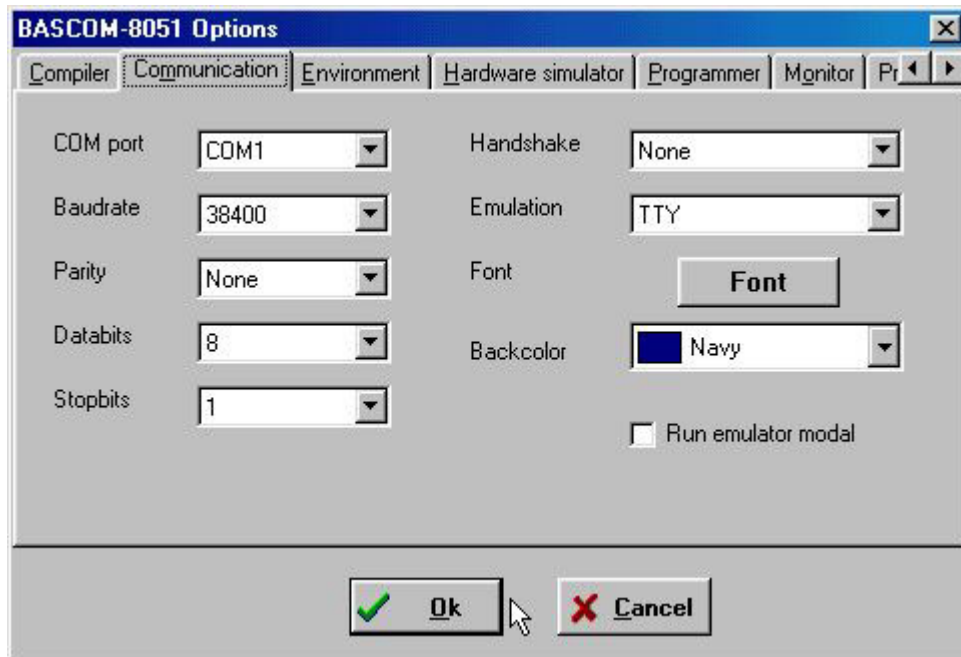


FIGURA 12: CONFIGURAZIONE EMULAZIONE TERMINALE BASCOM 8051

- 2) Tramite il menu *Options / Monitor*, selezionare la finestra Hex Mon e su questa impostare i settaggi necessari allo scaricamento di un file HEX all'FMO53, i relativi ritardi nella comunicazione, come illustrato in figura 13, e premere *OK*

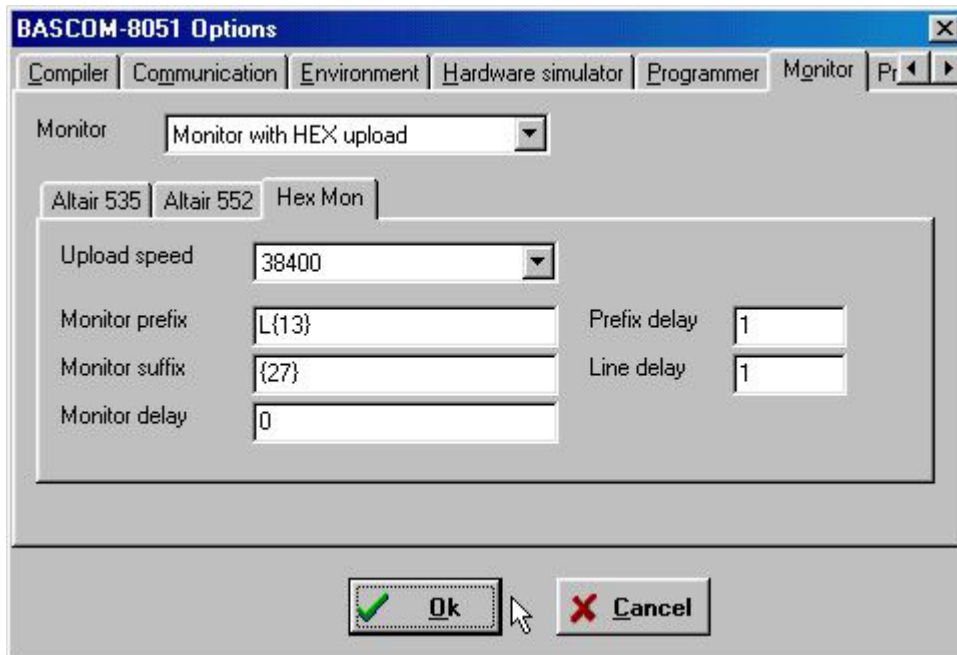


FIGURA 13: CONFIGURAZIONE SCARICAMENTO FILE CON BASCOM 8051

- 3) Scrivere il programma sorgente, o caricarlo da file, e compilarlo. Per maggiori informazioni su come compilare un programma **BASCOM 8051** adatto all'FMO53, si prega di leggere il paragrafo successivo. Si ricorda che ogni scheda della **grifo**® è fornita con un programma demo già predisposto per essere compilato con il **BASCOM 8051**, che può essere convenientemente usato in questa fase.

- 4) Entrare nell'emulatore di terminale premendo l'apposito pulsante sulla barra dei pulsanti oppure tramite il menu *Tools / Terminal emulator* oppure premendo la combinazione di tasti *Ctrl t*.
- 5) Alimentare o resettare la scheda di controllo: deve comparire la schermata iniziale di figura 26 o 27 con il prompt *. Ripetere i primi due punti di questa lista se tale videata non compare.
- 6) Attivare lo scaricamento del file HEX Intel generato al punto 3, con la voce *File / Upload File* della finestra *BASCOM-8051 Terminal Emulator* (vedere figura 14).

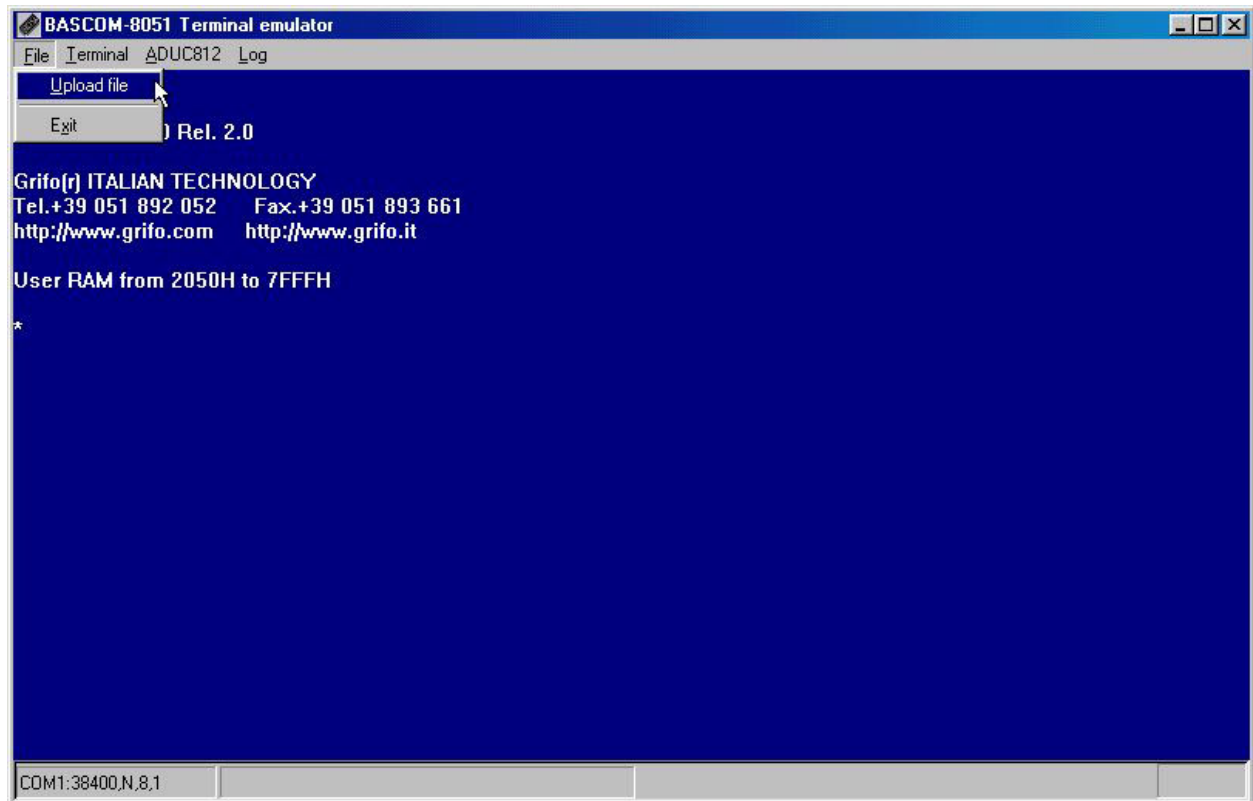


FIGURA 14: SCARICAMENTO FILE CON BASCOM 8051

- 7) Attendere la fine dello scaricamento evidenziato da un'apposita barra a scorrimento nell'angolo in basso a destra ed al termine verificare che appaia l'indicazione esadecimale con il conteggio del numero di bytes scaricati (coincide con la lunghezza del codice generato arrotondato per eccesso ad un multiplo di 16=10H).
- 8) Eseguire il programma scaricato digitando il comando **G** <Inizio codice utente>, dove **Inizio codice utente** corrisponde all'indirizzo di inizio del programma applicativo scaricato, in esadecimale (es. 2050).

DIRETTIVE DEL PROGRAMMA APPLICATIVO

Quando viene eseguita una compilazione di un sorgente **BASCOM 8051**, viene generato un file HEX Intel che contiene il codice macchina del microcontrollore utilizzato. Tramite alcune direttive del compilatore l'utente può configurare il programma affinché possa essere scaricato ed eseguito con l'**FMO53**, come abbondantemente descritto nel paragrafo CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE.

Le direttive di compilazione del **BASCOM 8051** possono essere definite in appositi menu dell'IDE od aggiunte direttamente nel sorgente. Tra queste due tecniche la seconda è indubbiamente vantaggiosa in quanto è prioritaria rispetto alla prima ed allo stesso tempo la rende immediatamente disponibile a qualsiasi programmatore e funzionante in qualsiasi condizione operativa.

In particolare:

Direttiva	Descrizione
<i>\$baud</i>	-> Indica il baud rate della comunicazione seriale del programma. Se il programma prevede una console seriale è consigliato settare tale console con lo stesso protocollo fisico dell' FMO53 , ovvero usare questa direttiva per impostare un baud rate = 38400.
<i>\$romstart</i>	-> Indica l'indirizzo del primo byte di codice del programma, ovvero coincide con l'indicazione Inizio codice utente , usata in questo manuale. Questa direttiva rialloca anche i vettori di risposta agli interrupts.
<i>\$ramstart</i>	-> Indica l'indirizzo del primo byte SRAM esterna disponibile come area dati del programma, ovvero coincide con l'indicazione Inizio dati utente , usata in questo manuale.
<i>\$ramsize</i>	-> Indica la dimensione della SRAM esterna utilizzabile come area dati e coincide con il valore di (Fine dati utente – Inizio dati utente).
<i>\$large</i>	-> Indica al compilatore di generare un codice che può essere maggiore di 2K, ovvero di indirizzare tutta la memoria disponibile.

Queste, ed altre direttive non direttamente utili per l'**FMO53**, sono dettagliatamente descritte nella documentazione del **BASCOM 8051**.

Come indicato nel precedente capitolo, i valori delle direttive cambiano a seconda delle condizioni operative e di seguito riportiamo alcuni tipici esempi di assegnazione:

- Programma in modalità DEBUG su seriale A:

```
$baud = 38400           ' Buad rate seriale hw
$romstart = &H2050      ' Inizio codice e vettori d'interrupt in SRAM
$ramstart = &H7000      ' Prevede 4K di area dati a partire da 7000H
$ramsize = &H0FFF
$large                  ' Dimensione codice > 2K
```

- Programma in modalità DEBUG su seriale B:

```
$romstart = &H2100      ' Inizio codice e vettori d'interrupt in SRAM
$ramstart = &H7000      ' Prevede 4K di area dati a partire da 7000H
$ramsize = &H0FFF
$large                  ' Dimensione codice > 2K
```

- Programma in modalità AUTORUN:

```
$baud = 38400           ' Buad rate seriale hw
$romstart = &H8050      ' Inizio codice e vettori d'interrupt in FLASH
$ramstart = &H7000      ' Prevede 4K di area dati a partire da 7000H
$ramsize = &H0FFF
$large                  ' Dimensione codice > 2K
```

UTILIZZO CON GET51

GET 51 è un emulatore di terminale intelligente compatibile con il protocollo ADDS-VIEWPOINT ed in grado di interfacciarsi direttamente con il famoso **MCS BASIC** e con l'**FMO53**, realizzato per il sistema operativo MS-DOS ma che può comunque essere usato anche con tutti i Windows fino alla versione ME. Per ulteriori informazioni sulle caratteristiche generali di questo prodotto si consiglia di esaminare il relativo foglio tecnico, il manuale d'uso oppure il completo aiuto in linea. Le informazioni relative all'integrazione con **FMO53** sono invece riportate nei seguenti paragrafi.

CONFIGURAZIONI PER COMUNICAZIONE SERIALE

Una delle funzionalità del **GET51** coincide con un emulatore terminale seriale che è completamente configurabile ed adattabile alle specifiche di funzionamento dell'**FMO53**. In aggiunta l'emulatore terminale è in grado di scaricare comodamente alla scheda, il codice di un generico programma applicativo.

In questo modo l'**FMO53** può ricevere dal **GET51** il comando di scaricamento di un file HEX Intel, il file stesso (generato da un altro ambiente di sviluppo) ed i caratteri di fine scaricamento, semplicemente attivando la voce di un apposito menù o premendo l'equivalente tasto rapido.

Per ottenere questo risultato bisogna usare il GET51 nel seguente modo:

- 1) Tramite il menu *Options / Serial Port...*, selezionare la porta seriale COMx usata sul PC, selezionare il baud rate di comunicazione dell'**FMO53**, come illustrato in figura 15, e premere *OK*.

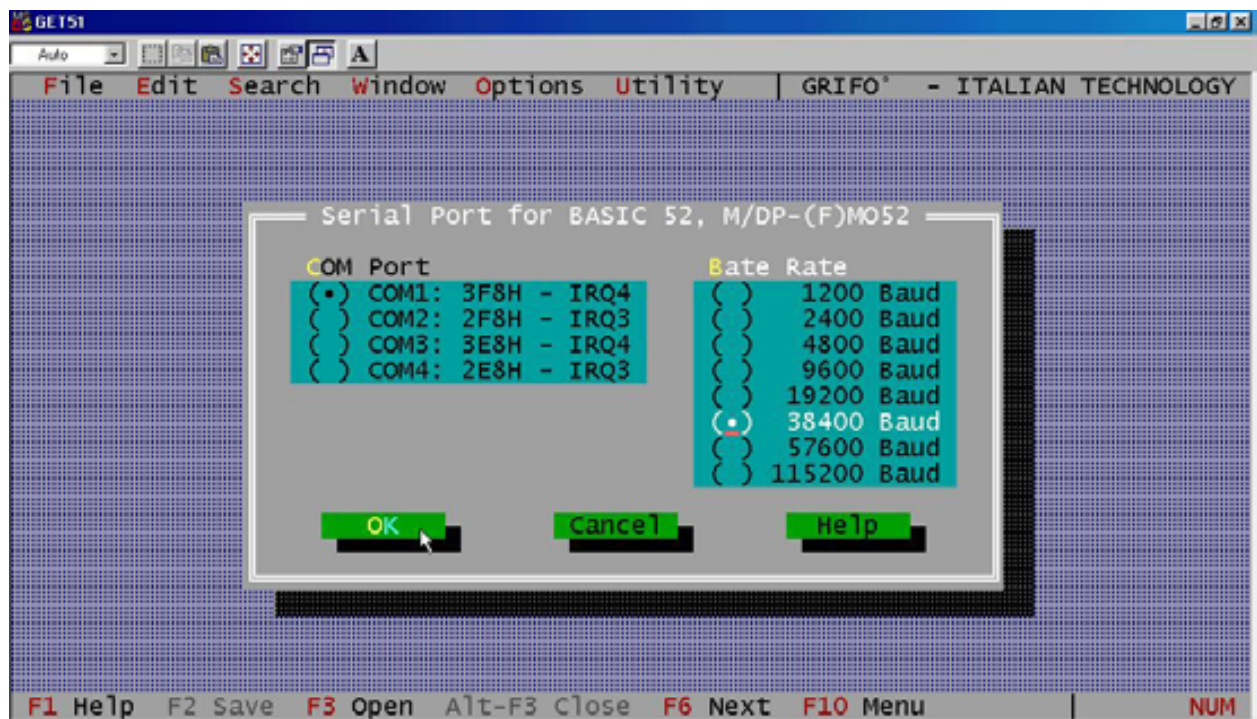


FIGURA 15: CONFIGURAZIONE SERIALE DEL GET51

- 2) Tramite il menu *Options / Set Terminal...*, selezionare il tipo di terminale *MD/P-(F)MO52* adatto all comunicazione con **FMO53** come illustrato in figura 16, e premere *OK*.

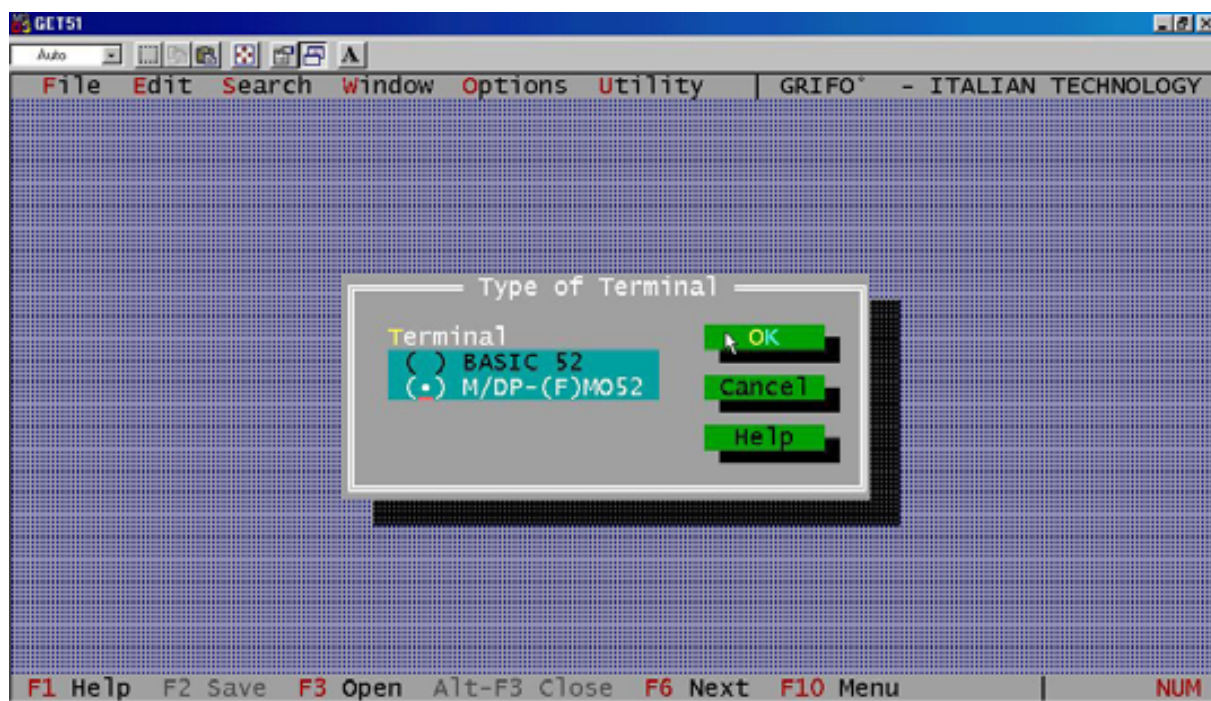


FIGURA 16: CONFIGURAZIONE TIPO TERMINALE CON GET51

- 3) Entrare nell'emulatore di terminale tramite il menu *Options / Terminal* oppure premendo la combinazione di tasti *Alt t*.
- 4) Alimentare o resettare la scheda di controllo: deve comparire la schermata iniziale di figura 26 o 27 con il prompt *. Ripetere i primi tre punti di questa lista se tale videata non compare.
- 5) Attivare lo scaricamento del file HEX Intel che deve essere salvato su uno dei dischi del PC di sviluppo, con la voce *Utility / Transmit File...* o, più semplicemente, con la pressione del tasto *F8*. La finestra che appare (vedere figura 17) permette di cercare il file desiderato su tutti i dischi del PC, di selezionarlo e di confermarlo premendo *INVIO* od *OK*.

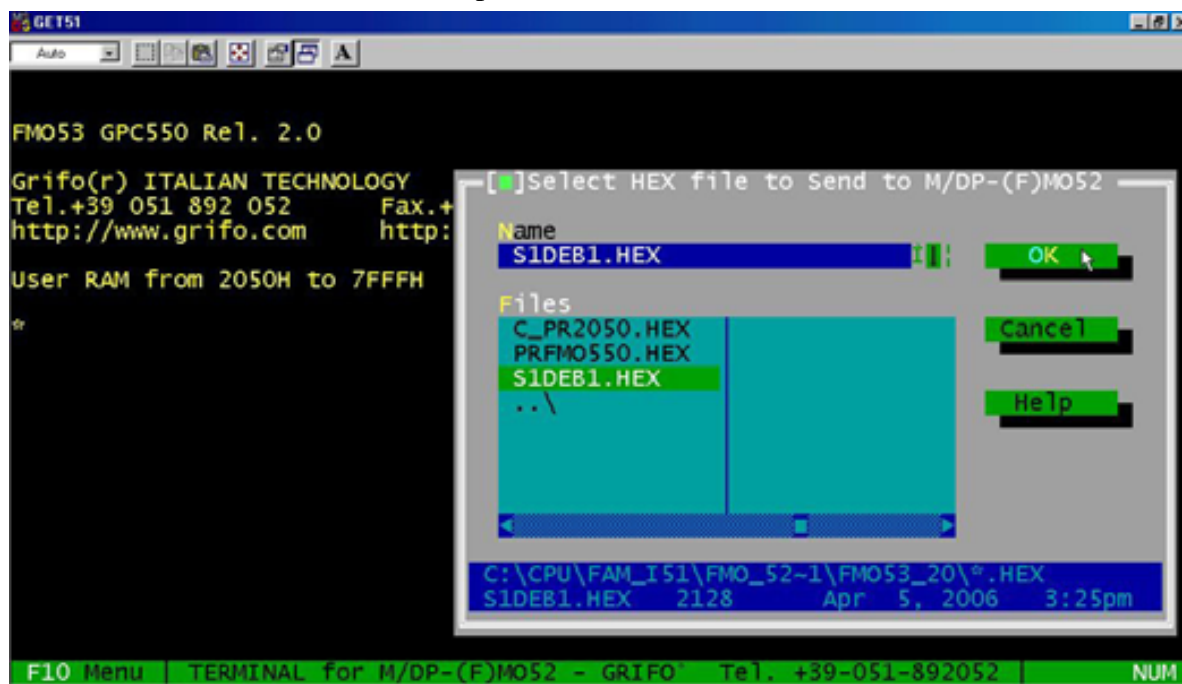


FIGURA 17: SCARICAMENTO FILE CON GET51

- 6) Attendere la fine dello scaricamento evidenziato da un'apposita barra a scorrimento nella finestra che compare ed al termine verificare che appaia l'indicazione esadecimale con il conteggio del numero di bytes scaricati (coincide con la lunghezza del codice generato arrotondato per eccesso ad un multiplo di 16=10H).
- 7) Eseguire il programma scaricato digitando il comando *G <Inizio codice utente>*, dove **Inizio codice utente** corrisponde con l'indirizzo di inizio del programma applicativo scaricato, in esadecimale (es. 2050).

UTILIZZO CON HYPERTERMINAL

HYPERTERMINAL è un famoso programma di comunicazione che supporta anche la funzione di emulatore terminale compatibile con l'**FMO53**. E' uno degli accessori forniti con il sistema operativo Windows e per questo disponibile in tutte le sue versioni. Per ulteriori informazioni sulle caratteristiche generali di questo prodotto si consiglia di esaminare la relativa guida in linea. Le informazioni relative all'integrazione con **FMO53** sono invece riportate nei seguenti paragrafi.

CONFIGURAZIONI PER COMUNICAZIONE SERIALE

Tra le molteplici funzionalità di **HYPERTERMINAL** si ricorda quella di emulatore terminale seriale che è completamente configurabile ed adattabile alle specifiche di funzionamento dell'**FMO53**. In aggiunta il programma è in grado di scaricare comodamente alla scheda, il codice di un generico programma applicativo.

In questo modo l'**FMO53** può ricevere da **HYPERTERMINAL** il comando di scaricamento di un file HEX Intel, il file stesso (generato da un altro ambiente di sviluppo) ed i caratteri di fine scaricamento, in una modalità semiautomatica.

Per ottenere questo risultato bisogna usare **HYPERTERMINAL** nel seguente modo:

- 1) Lanciare il programma dalla sua cartella d'installazione che normalmente è: *Start / Programmi / Accessori / Comunicazioni / Hyper Terminal*. A questo punto compare una finestra di *Descrizione della connessione* in cui deve essere inserito il nome **FMO53**, come indicato nella figura 18, ed infine preme il pulsante *OK*.



FIGURA 18: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (1 DI 4)

2) Nella successiva finestra *Connetti a* provvedere a selezionare l'opzione di collegamento diretto alla COM usata sul PC, in corrispondenza del campo *Connetti*, come indicato in figura 19, e confermare con il pulsante *OK*.



FIGURA 19: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (2 DI 4)

3) Nella finestra *Proprietà COMx* che compare, configurare i parametri del protocollo fisico di comunicazione dell'**FM053**, disabilitare gli handshake hardware, come illustrato in figura 20, e premere *OK*.

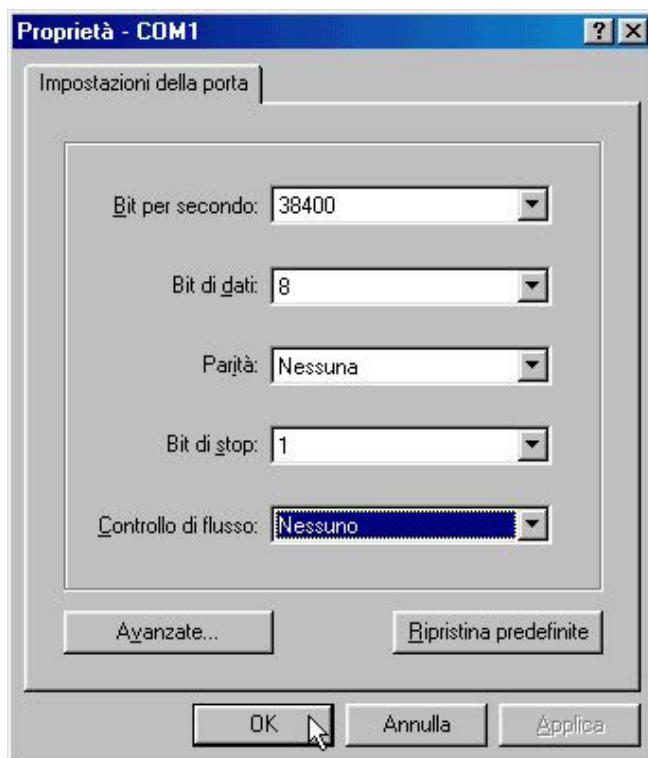


FIGURA 20: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (3 DI 4)

4) Salvare le configurazioni effettuate in un file in modo da poterle richiamare velocemente durante tutti i futuri usi dell'**FM053** con **HYPERTERMINAL**, tramite l'opzione *File / Salva col nome....* Si consiglia ad esempio di chiamare *FM053.HT* tale file di configurazione (come illustrato in figura 21), e di usarlo per ogni esecuzione, con un semplice doppio click sulla sua icona.

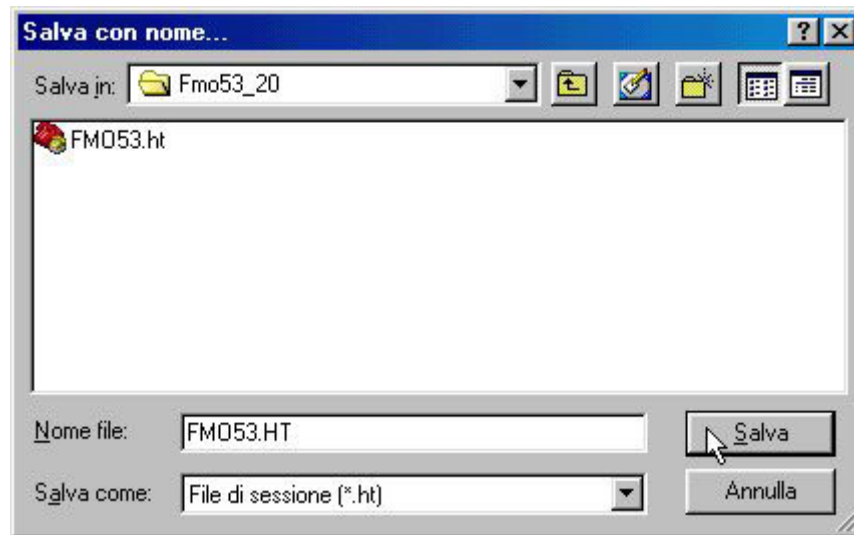


FIGURA 21: CONFIGURAZIONE COMUNICAZIONE CON HYPERTERMINAL (4 DI 4)

- 5) Alimentare o resettare la scheda di controllo: deve comparire la schermata iniziale di figura 26 o 27 con il prompt *. Ripetere i primi tre punti di questa lista se tale videata non compare.
- 6) Fornire il comando di scaricamento file premendo in sequenza i tasti *L* ed *INVIO*.
- 7) Attivare lo scaricamento del file HEX Intel che deve essere salvato su uno dei dischi del PC di sviluppo, con la voce *Trasferisci / Invia file di testo...*. La finestra che appare (vedere figura 23) permette di cercare il file desiderato su tutti i dischi del PC, di selezionarlo e di confermarlo premendo *INVIO* od *OK*.

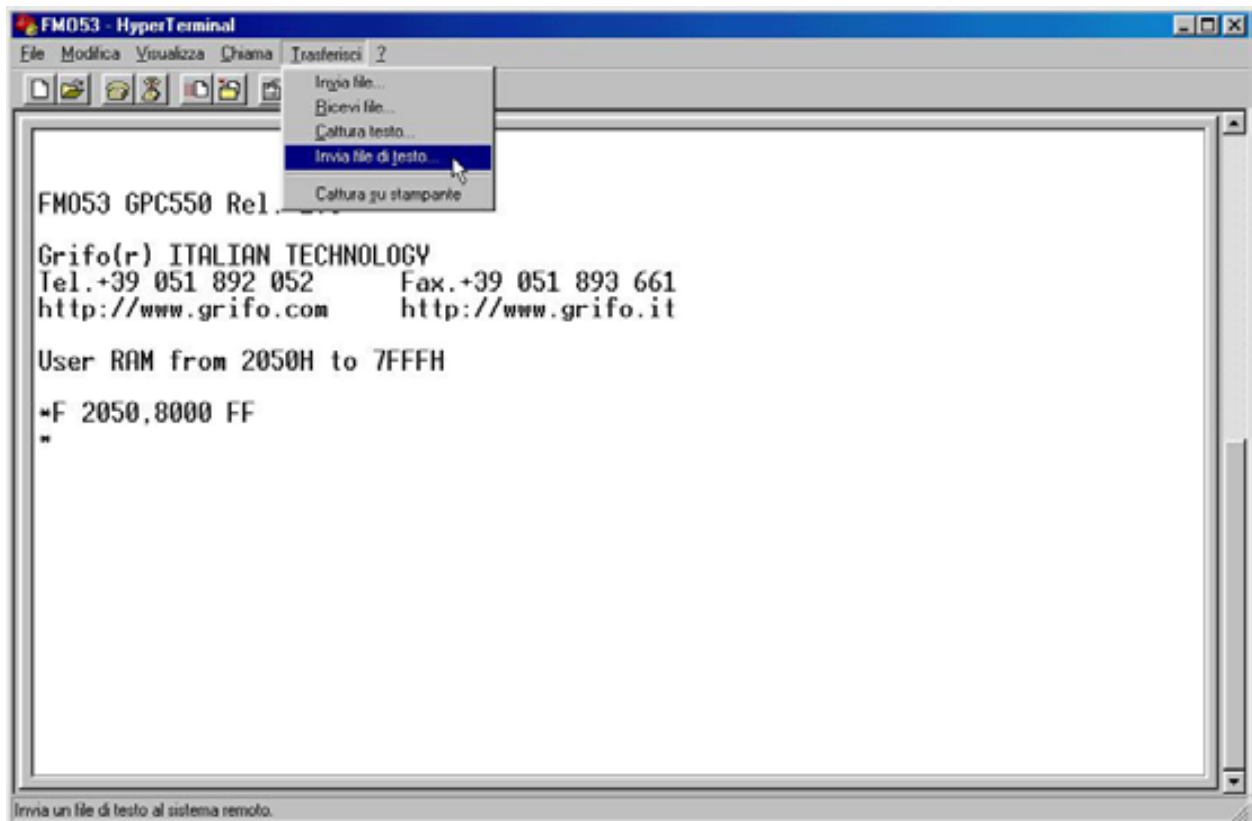


FIGURA 22: SCARICAMENTO FILE CON HYPERTERMINAL (1 DI 2)

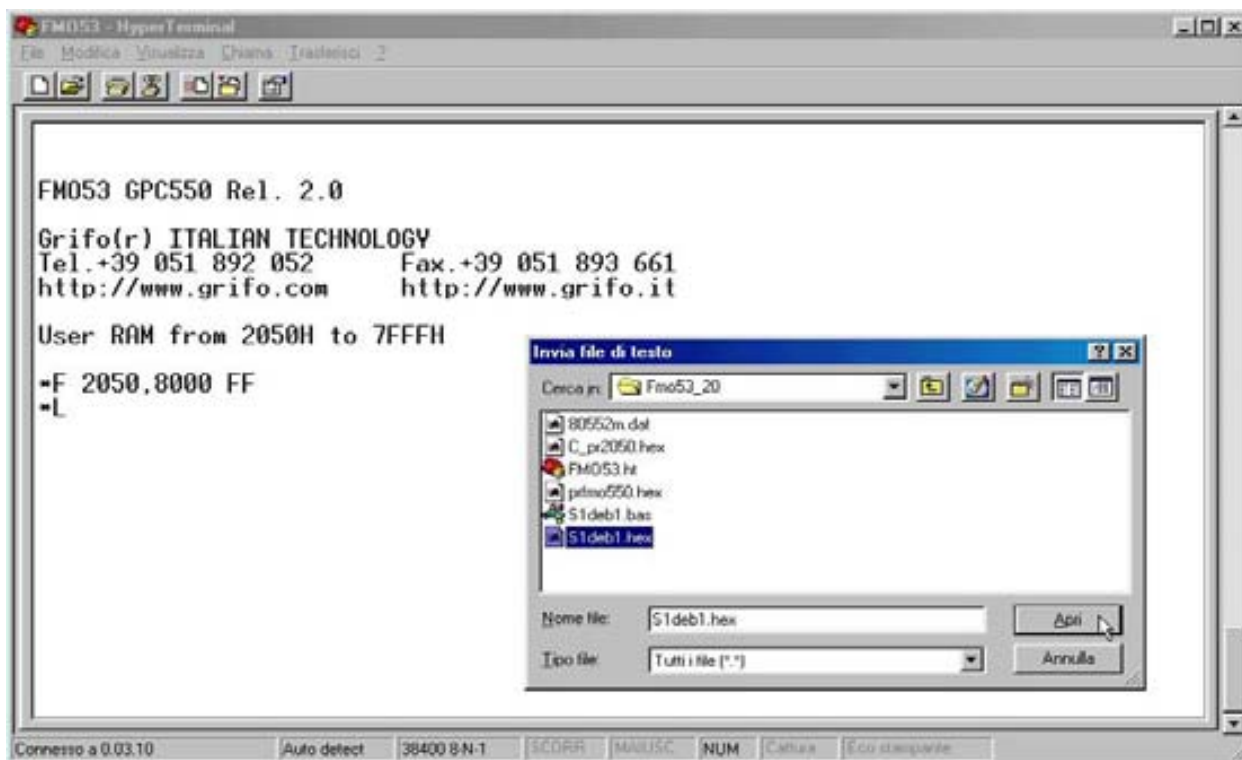


FIGURA 23: SCARICAMENTO FILE CON HYPERTERMINAL (2 DI 2)

- 8) Attendere la fine dello scaricamento evidenziato dalla comparsa dell'indicazione esadecimale con il conteggio del numero di bytes scaricati (coincide con la lunghezza del codice generato arrotondato per eccesso ad un multiplo di 16=10H). Durante questa attesa l'utente deve fare attenzione a non premere alcun tasto sulla tastiera, infatti questo verrebbe trasmesso sulla seriale andando a modificare il file contemporaneamente trasmesso e generando un errore.
- 9) Eseguire il programma scaricato digitando il comando **G** <*Inizio codice utente*>, dove **Inizio codice utente** corrisponde con l'indirizzo di inizio del programma applicativo scaricato in esadecimale (es. 2050).

UTILIZZO CON μ C/51

μ C/51 è un cross compilatore C, economico, efficiente, facile da usare, provvisto di un completo IDE (Integrated Development Environment = ambiente di sviluppo integrato), che gira sotto qualunque versione di Windows e genera codice eseguibile per microcontrollori della famiglia 'I51. Per ulteriori informazioni sulle caratteristiche generali di questo prodotto si consiglia di esaminare il relativo foglio tecnico ed il manuale d'uso. Le informazioni relative all'integrazione con **FM053** sono invece riportate nei seguenti paragrafi.

CONFIGURAZIONI PER COMUNICAZIONE SERIALE

Tra il gruppo di programmi installati con il $\mu\text{C}/51$ nessuno si presta a gestire la comunicazione seriale con la scheda di controllo che esegue l'**FMO53**. Per questa ragione si consiglia di usare uno dei programmi di sola emulazione terminale (come **GET51** od **HYPERTERMINAL**) descritti nei paragrafi precedenti. Eventualmente è possibile integrare il lancio del programma di comunicazione prescelto nell'IDE del *Jen's File Editor (JFE)*, quando si usano i file di workspace generati dal programma di utilità *MakeWiz*. Per ottenere questa integrazione è sufficiente leggere il manuale utente del $\mu\text{C}/51$ e modificare il file *DL.BAT*, in modo che lanci il programma di comunicazione che si intende usare.

Ad esempio se si decide di utilizzare il programma **HYPERTERMINAL** si dovranno prima effettuare tutte le operazioni descritte nel paragrafo **UTILIZZO CON HYPERTERMINAL** e poi inserire le righe di comando seguenti nel file *DL.BAT*:

```
@ECHO OFF
REM Attiva emulazione terminale con HYPERTERMINAL configurato
REM da file FMO53.HT
C:\PROGRA~1\ACCESS~1\HYPERT~1\HYPERTRM.EXE FMO53.HT
ECHO ON
```

DIRETTIVE DEL PROGRAMMA APPLICATIVO

Quando viene eseguita una compilazione di un sorgente $\mu\text{C}/51$, viene generato un file HEX Intel che contiene il codice macchina del microcontrollore utilizzato. Tramite alcune direttive del compilatore l'utente può configurare il programma affinché possa essere scaricato ed eseguito con l'**FMO53**, come abbondantemente descritto nel paragrafo **CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE**.

Le direttive di compilazione del $\mu\text{C}/51$ devono essere definite in appositi file di progetto .MAK che istruiscono il compilatore nel suo processo di trasformazione. Tali file di progetto possono essere convenientemente realizzati con l'apposito programma di utilità *MakeWiz*, che, tramite una serie di finestre, consente la definizione di tutte le informazioni necessarie come: nome file sorgenti, modello di memoria, ottimizzazioni, file generati, indirizzi di allocazione, librerie usate, ecc.

Di queste numerose direttive solo tre riguardano l'**FMO53**, ovvero:

Direttiva	Descrizione
<i>Linker / Rom Start (Hex)</i>	-> Indica l'indirizzo del primo byte di codice del programma, ovvero coincide con l'indicazione Inizio codice utente , usata in questo manuale. Questa direttiva rialloca anche i vettori di risposta agli interrupts.
<i>Linker / External Ram Start (Hex)</i>	-> Indica l'indirizzo del primo byte SRAM esterna disponibile come area dati del programma, ovvero coincide con l'indicazione Inizio dati utente , usata in questo manuale.
<i>Misc / Gen. HEX-File</i>	-> Indica al compilatore di generare il codice del programma applicativo nel formato HEX Intel, accettato dall' FMO53

Queste, ed altre direttive non direttamente utili per l'**FMO53**, sono dettagliatamente descritte nella documentazione del $\mu\text{C}/51$.

Come indicato nel precedente capitolo, i valori delle direttive cambiano a seconda delle condizioni operative e di seguito riportiamo alcuni tipici esempi di assegnazione:

- Programma in modalità DEBUG su seriale A:

Rom Start (Hex) = 2050

External Ram Start (Hex) = 7000

- Programma in modalità DEBUG su seriale B:

Rom Start (Hex) = 2100

External Ram Start (Hex) = 7000

- Programma in modalità AUTORUN:

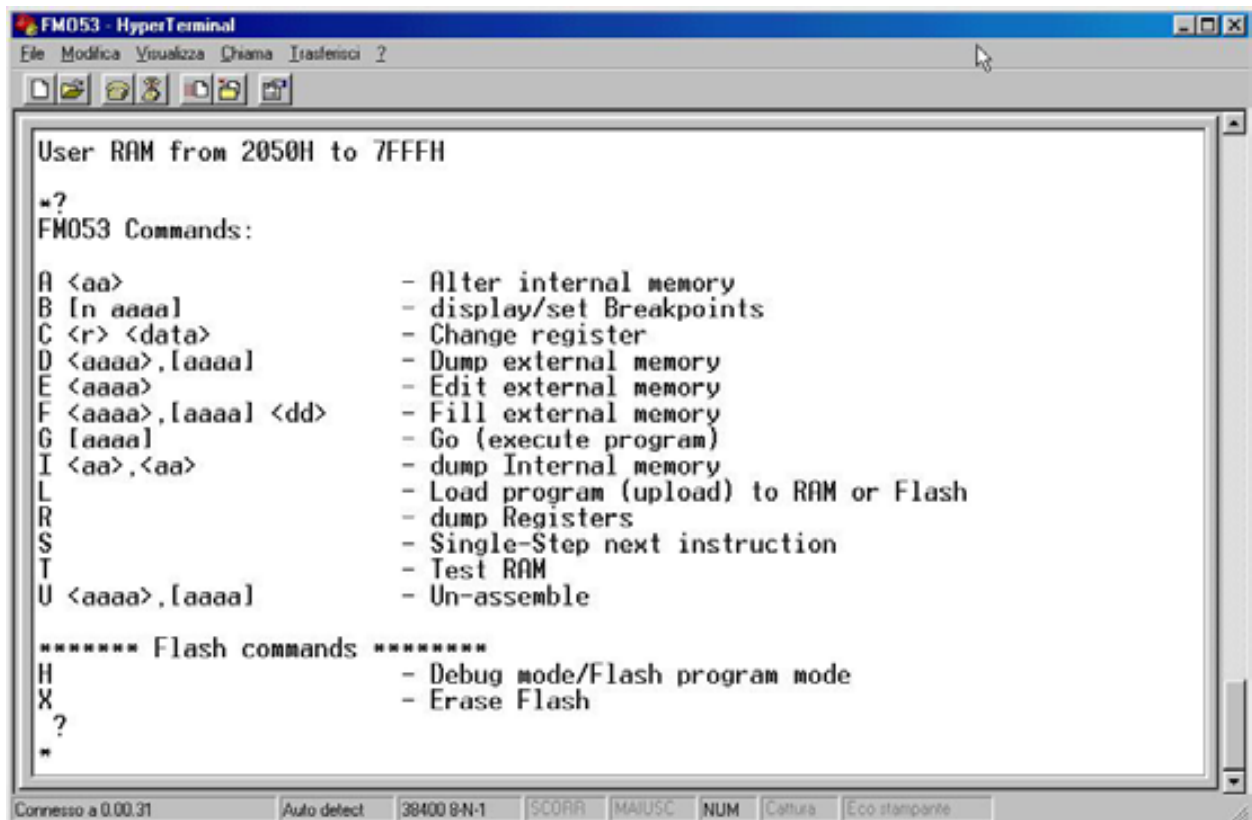
Rom Start (Hex) = 8050

External Ram Start (Hex) = 7000

COMANDI

Tutte le funzioni svolte dall'**FMO53** sono attivate da una serie di comandi, che consentono all'utente di interagire con la scheda di controllo ed ottenere gli obiettivi prefissati. La completezza dei comandi è una delle caratteristiche fondamentali del pacchetto ed, in fase progettuale, ci si è posti l'obiettivo di ottenere un'operatività paragonabile a quella di un "In Circuit Emulator" hardware. Al seguito della partenza in modalità DEBUG, l'**FMO53**, rappresenta il carattere di pronto * a capo riga e rimane in attesa di comandi dall'utente. Molti comandi sono caratterizzati da una riga di comando, su cui vengono passati i parametri necessari, per svolgere l'azione corrispondente. L'intera riga di comando deve essere quindi digitata sulla tastiera del PC di sviluppo e sul monitor di quest'ultimo vengono rappresentati gli eventuali risultati del comando eseguito.

Al fine di avere la massima quantità di informazioni su una videata del PC di sviluppo, l'**FMO53** è stato progettato per ridurre al minimo le linee di rappresentazione. Tutti i comandi che rappresentano solo una riga di informazioni (come **B** ed **S**) lo fanno sulla stessa riga su cui è stato fornito il comando (alla sua destra). Per i comandi che invece visualizzano grosse quantità di dati (come **D**, **U**, ecc.) premendo un qualsiasi tasto durante la rappresentazione, questa si sospenderà. A questo punto ogni successiva pressione dello **SPAZIO** rappresenterà una ulteriore riga, mentre la pressione di **INVIO** riprenderà la rappresentazione a piena velocità delle righe rimanenti; infine il tasto **ESC** può essere utilizzato per interrompere definitivamente la rappresentazione del comando.



```

FMO53 - HyperTerminal
File Modifica Visualizza Chiama Trasferisci ?
[Icons]
User RAM from 2050H to 7FFFH
*?
FMO53 Commands:

A <aa>                - Alter internal memory
B [n aaaa]            - display/set Breakpoints
C <r> <data>           - Change register
D <aaaa>,[aaaa]       - Dump external memory
E <aaaa>              - Edit external memory
F <aaaa>,[aaaa] <dd>  - Fill external memory
G [aaaa]              - Go (execute program)
I <aa>,<aa>            - dump Internal memory
L                     - Load program (upload) to RAM or Flash
R                     - dump Registers
S                     - Single-Step next instruction
T                     - Test RAM
U <aaaa>,[aaaa]       - Un-assemble

***** Flash commands *****
H                     - Debug mode/Flash program mode
X                     - Erase Flash
?
*

```

Connesso a 0.00.31 Auto detect 38400 8-N-1 SCORR MAIUSC NUM Cattura Eco stampante

FIGURA 24: ELENCO COMANDI

Nei paragrafi seguenti vengono descritti tutti i comandi dell'**FMO53** con le relative indicazioni sugli eventuali parametri d'ingresso e risultati rappresentati. In questi paragrafi si assume che il lettore conosca il microcontrollore usato e si usano quindi i suoi termini tipici per indicare le sezioni di interesse.

ALTERA MEMORIA INTERNA

A <indirizzo>

Altera il contenuto della RAM interna del microcontrollore a partire dall'*indirizzo* specificato. Una volta riconosciuto il comando l'**FMO53** presenta l'*indirizzo* specificato ed il suo attuale contenuto: a questo punto si possono digitare due cifre esadecimali per cambiare il valore, lo *SPAZIO* per avanzare alla prossima locazione, il *BACKSPACE* per ritornare alla locazione precedente o *INVIO* per terminare il comando di alterazione.

SETTA E VISUALIZZA PUNTI D'ARRESTO

B <n. punto d'arresto> <indirizzo>

Setta un punto d'arresto (breakpoint) all'*indirizzo* specificato. L'*indirizzo* deve essere inserito con 4 cifre esadecimali e se quello specificato è 0000, il punto d'arresto è rimosso. L'**FMO53** prevede fino a quattro punti d'arresto diversi, numerati da 0 a 3.

Se è premuto lo *SPAZIO* al posto del *n. punto d'arresto*, sono visualizzati gli indirizzi degli attuali punti d'arresto, già impostati.

Note:

La famiglia 8051 di microprocessori non ha una istruzione di salto che occupi un singolo byte e quindi i punti d'interruzione non possono essere gestiti con le modalità classiche. I punti d'arresto sono quindi implementati inserendo nel codice un'istruzione LCALL a seguito dell'esecuzione con il comando **G** (GO) e ripristinando il codice originario, quando il punto d'arresto viene raggiunto. L'istruzione LCALL occupa tre byte di memoria codice e questo provoca i seguenti limiti:

- Bisogna fare attenzione a non settare punti d'arresto in posizioni i cui due indirizzi successivi vengono chiamati da altre parti del programma. Per esempio, se viene settato un punto d'interruzione all'*indirizzo* 2234, nel programma non ci devono essere istruzioni che saltano agli indirizzi 2235 o 2236 (un salto all'*indirizzo* 2234 è invece corretto).
- Non si possono settare punti d'arresto che vanno sui tre byte occupati da un altro punto d'interruzione. Se si tenta questo settaggio viene rappresentato il messaggio d'errore *Breakpoint conflict*.
- Non si possono fornire comandi di **G** (GO) ad un *indirizzo* che contiene un punto d'arresto; anche in questo caso viene visualizzato il messaggio d'errore *Breakpoint conflict*. Si cade frequentemente in questo limite quando si vuole riprendere l'esecuzione a seguito del raggiungimento di un punto d'interruzione; in questo caso si deve invece prima rimuovere il punto d'arresto raggiunto oppure usare il comando **S** (STEP) per avanzare il puntatore programma (PC) fino a sorpassare le locazioni occupate dal punto d'arresto.

MODIFICA REGISTRI

C <nome registro> <valore>

Cambia i valori dei registri interni dell'8051. Il *nome registro* da modificare coincide con una singola lettera ed il *valore* può essere a 8 o 16 bits, come sotto descritto:

A	->	Registro A (accumulatore)	->	valore a 8 bit.
B	->	Registro B (registro generico)	->	valore a 8 bit.
D	->	Registro DPTR (puntatore dati)	->	valore a 16 bit.
S	->	Registro SP (puntatore stack)	->	valore a 8 bit.
P	->	Registro PC (puntatore programma)	->	valore a 16 bit.
W	->	Registro PSW (registro di stato)	->	> valore a 8 bit.
0-7	->	Registri R0-R7 (banco registri attuale)	->	valore a 8 bit.

Il *valore* da assegnare deve essere inserito in esadecimale a 2 cifre quando il registro è ad 8 bit, oppure a 4 cifre se è a 16 bit.

VISUALIZZA MEMORIA DATI ESTERNA

D <indirizzo inizio> <indirizzo fine>

Mostra il contenuto della memoria dati esterna compresa tra gli indirizzi specificati, in formato esadecimale ed ASCII. Se come indirizzo di fine è premuto lo *SPAZIO* allora si assume un indirizzo di fine della memoria = FFFFH.

MODIFICA MEMORIA DATI ESTERNA

E <indirizzo>

Modifica il contenuto dell'area dati esterna del microcontrollore a partire dall'*indirizzo* specificato. Una volta riconosciuto il comando l'**FMO53** presenta l'indirizzo specificato ed il suo attuale contenuto: a questo punto si possono digitare due cifre esadecimali per cambiare il valore, lo *SPAZIO* per avanzare alla prossima locazione, il *BACKSPACE* per ritornare alla locazione precedente o *INVIO* per terminare il comando di modifica.

RIEMPE MEMORIA DATI ESTERNA

F <indirizzo inizio> <indirizzo fine> <valore>

Riempe la memoria dati esterna dall'*indirizzo inizio* fino all'*indirizzo fine* con il valore HEX specificato. Ricordare che il comando non chiede conferme ed una volta inserito la seconda cifra del valore con cui riempire la memoria, il precedente contenuto è definitivamente perso.

ESEGUE A VELOCITÀ REALE

G <indirizzo>

Esegue il programma salvato nella memoria codice a partire dall'indirizzo specificato. Se uno *SPAZIO* e' premuto al posto dell'indirizzo, l'esecuzione inizia all'indirizzo attualmente contenuto nel registro puntatore programma (PC).

ABILITA COMANDI PER FLASH

H

Abilita l'utilizzo dei comandi dedicati all'uso della FLASH. Questa modalità viene indicata dal carattere di pronto, rappresentato ad inizio riga, che si differenzia come segue:

- * -> Debug mode -> i comandi per FLASH non sono abilitati
- & -> FLASH program mode -> i comandi per FLASH sono abilitati

In corrispondenza del riconoscimento di questo comando l'**FMO53** commuta dalla modalità attuale alla rimanente, rappresentando nell'ordine la modalità selezionata, la sua organizzazione delle memorie ed infine il carattere di pronto corrispondente, come indicato in figura 25.

Si ricorda che l'abilitazione dei comandi per FLASH modifica anche il comportamento del comando *L* (UPLOAD), come indicato nel corrispondente paragrafo.

VISUALIZZA MEMORIA INTERNA

I <indirizzo inizio> <indirizzo fine>

Mostra il contenuto della memoria RAM interna del microcontrollore compresa tra i due indirizzi specificati, in formato esadecimale.


```

FM053 GPC550 Rel. 2.0
Grifo(r) ITALIAN TECHNOLOGY
Tel. +39 051 892 052 Fax. +39 051 893 661
http://www.grifo.com http://www.grifo.it

User RAM from 2050H to 7FFFH

*H
Flash program mode
User Code = 8050H to FEFFH - User Data = E000H to FEFFH
Now you must upload file/s to program Flash

&H
Debug mode
User RAM from 2050H to 7FFFH

*_

```

FIGURA 25: COMANDO ABILITAZIONE COMANDI PER FLASH

CARICA FILE

L

Attiva la ricezione di un file in formato HEX Intel od S Motorola dal PC di sviluppo. Il comando termina in corrispondenza della ricezione delle sequenze :00 o S9, che normalmente coincidono con i record di fine file. Se viene fornito questo comando involontariamente, l'utente può ritornare al modo comandi digitando direttamente una delle due sequenze descritte.

Il funzionamento di questo comando varia concordemente con l'abilitazione dei comandi per FLASH, infatti se il carattere di pronto é * (comandi per FLASH non abilitati) il codice scaricato viene salvato nella SRAM della scheda di controllo (modalità DEBUG), mentre se il carattere é & (comandi per FLASH abilitati) il codice scaricato viene salvato nella FLASH. In quest'ultimo caso se lo scaricamento del file e la programmazione della FLASH avvengono senza errori, al termine viene impostata la condizione di AUTORUN sempre sulla FLASH. In questo modo alla successiva accensione il programma applicativo utente scaricato verrà automaticamente eseguito, come indicato nel paragrafo SELEZIONE MODO OPERATIVO.

Al termine dello scaricamento avvenuto con successo, l'FM053 rappresenta il numero di Bytes salvati nella memoria della scheda, arrotondato per eccesso ad un multiplo di 16=10H. Viceversa in caso di problemi nel file scaricato (record non valido, dato non consentiti, indirizzo non ammessi, CRC errato, ecc.) o nella programmazione della FLASH viene rappresentato un adeguato messaggio di errore.

Per dettagli sulle informazioni rappresentate dal comando *L*, vedere le figure 28 e 29 di questo manuale.

VISUALIZZA REGISTRI

R

Mostra i valori attuali dei 14 registri del microcontrollore 8051 (A, B, DPTR, SP, PC, PSW e R0÷R7) in esadecimale. Questo comando, così come tutti quelli di visualizzazione, si dimostra particolarmente utile nella fase di ricerca problemi del programma applicativo, infatti consente di verificare lo stato di esecuzione, individuare gli eventuali valori diversi e quindi trovare la causa del malfunzionamento.

ESEGUE A SINGOLO PASSO

S

Esegue una singola istruzione memorizzata all'indirizzo attualmente salvato nel puntatore programma (PC). L'**FMO53** rappresenta inoltre il disassemblato dell'istruzione eseguita sul PC di sviluppo ed effettua automaticamente l'aggiornamento del PC in modo che punti all'istruzione successiva. L'uso di questo comando comporta l'uso del Timer1 e del suo interrupt come descritto nel paragrafo **RISORSE UTILIZZATE**.

Note:

Con i microcontrollori 80c320 Dallas e 80c552, vi sono rari casi in cui questo comando esegue 2 istruzioni invece di una, fermandosi all'istruzione che segue, quella da cui è stato lanciato il comando.

TESTA SRAM DELLA SCHEDA

T

Effettua un test completo della SRAM a partire dall'indirizzo di inizio (2050H in caso di uso seriale A della scheda oppure 2100 in caso di seriale B) fino a 7FFFH. Questo comando altera completamente il contenuto della SRAM e per garantire una condizione certa, una volta eseguito il comando *T*, si consiglia di formattare la SRAM con il valore 00H, tramite il comando *F 2050 7FFF 00*.

La funzione principale di questo comando è quella di verificare il corretto funzionamento della SRAM di base della scheda di controllo ed allo stesso tempo controllare che la scheda sia correttamente configurata. Infatti con un mappaggio o configurazione errata delle memorie l'**FMO53** può comunque partire ma il comando di test non viene superato.

DISASSEMBLA CODICE

U <indirizzo fine> <indirizzo fine>

Disassembla la memoria programma compresa tra i due indirizzi specificati. Il comando rappresenta l'indirizzo, i byte di codice, il mnemonico e gli operandi di ogni istruzione, usando sempre la notazione esadecimale per tutti i valori numerici.

Se come *indirizzo fine* e' premuto lo *SPAZIO* allora si assume un indirizzo di fine della memoria = FFFFH.

CANCELLA FLASH

X

Effettua la cancellazione totale del contenuto della FLASH. Una volta riconosciuto il comando **FMO53** richiede una conferma all'utente in modo da assicurare che la FLASH non venga cancellata involontariamente. Inoltre, sempre per aumentare la sicurezza, questo comando può essere eseguito solo se sono stati preventivamente abilitati i comandi per FLASH, con il comando **H**.

Si sottolinea che la cancellazione della FLASH é totale, quindi una volta eseguito il comando sono definitivamente persi sia il programma applicativo che i dati su FLASH; per maggiori informazioni si veda il paragrafo GESTIONE FLASH EPROM.

VISUALIZZA COMANDI

?

Mostre un breve sommario dei comandi disponibili completi dei parametri eventualmente richiesti (vedere figura 25).

COME INIZIARE

In questo capitolo vengono descritte quali sono le operazioni da effettuare per un primo utilizzo del pacchetto software **FMO53**. In particolare viene riportata la giusta sequenza di operazioni che l'utente deve effettuare prima per configurare e poi per svolgere le prime fondamentali operazioni, con le schede di controllo **grifo®**.

Nel capitolo si trovano molti riferimenti a precedenti paragrafi: per una corretta comprensione di tutte le informazioni, si consiglia di eseguire le operazioni di seguito descritte, solo dopo aver letto tutto il manuale.

- 1) Leggere tutta la documentazione ricevuta (manuale tecnico della scheda di controllo, manuale d'uso dell'**FMO53** e manuale d'uso dell'ambiente di sviluppo).
- 2) Stabilire quale linea seriale deve essere utilizzata sulla scheda remota ed effettuarne la selezione, come indicato nel paragrafo COMUNICAZIONE SERIALE.
- 3) Configurare la scheda di controllo a seconda delle memorie montate, seguendo le indicazioni del paragrafo CONFIGURAZIONE SCHEDA.
- 4) Impostare il selettore della scheda di controllo in modo da selezionare la modalità DEBUG, così come descritto nel paragrafo SELEZIONE MODO OPERATIVO.
- 5) Accendere il PC di sviluppo e selezionare su questo la linea seriale da utilizzare. Qualora tale seriale provenga da un convertitore (ad esempio interfacce USB <-> RS 232), effettuare le opportune operazioni indicate nella documentazione dello stesso prodotto, e verificarne il corretto funzionamento.
- 6) Effettuare il collegamento seriale seguendo le indicazioni del paragrafo CAVO DI COMUNICAZIONE SERIALE.
- 7) Installare sul PC l'ambiente di sviluppo prescelto e, se questo include una propria modalità di emulazione terminale, provvedere a configurarla per la comunicazione con **FMO53**. Se invece manca predisporre un programma di comunicazione esterno, come ad esempio **HYPERTERMINAL** di Windows. Nella configurazione si deve naturalmente selezionare la linea di comunicazione del PC, collegata al punto 6. Per dettagli sull'integrazione dell'**FMO53** descritta, fare riferimento agli appositi paragrafi COMUNICAZIONE SERIALE e CONFIGURAZIONI PER COMUNICAZIONE SERIALE.
- 8) Attivare l'emulazione terminale nel programma configurato al punto 7.
- 9) Fornire l'alimentazione alla scheda di controllo e verificare che sul monitor del PC di sviluppo compaia il messaggio di presentazione dell'**FMO53**. Quest'ultimo riporta nell'ordine: il nome della scheda usata, la versione dell'**FMO53**, le generalità della **grifo®**, gli indirizzi di inizio e fine della SRAM di base a disposizione dell'utente ed infine il carattere di pronto *. Da ricordare che l'indirizzo di inizio della SRAM utente, varia a seconda della linea seriale usata sulla scheda, come indicato nelle figure 26 e 27.

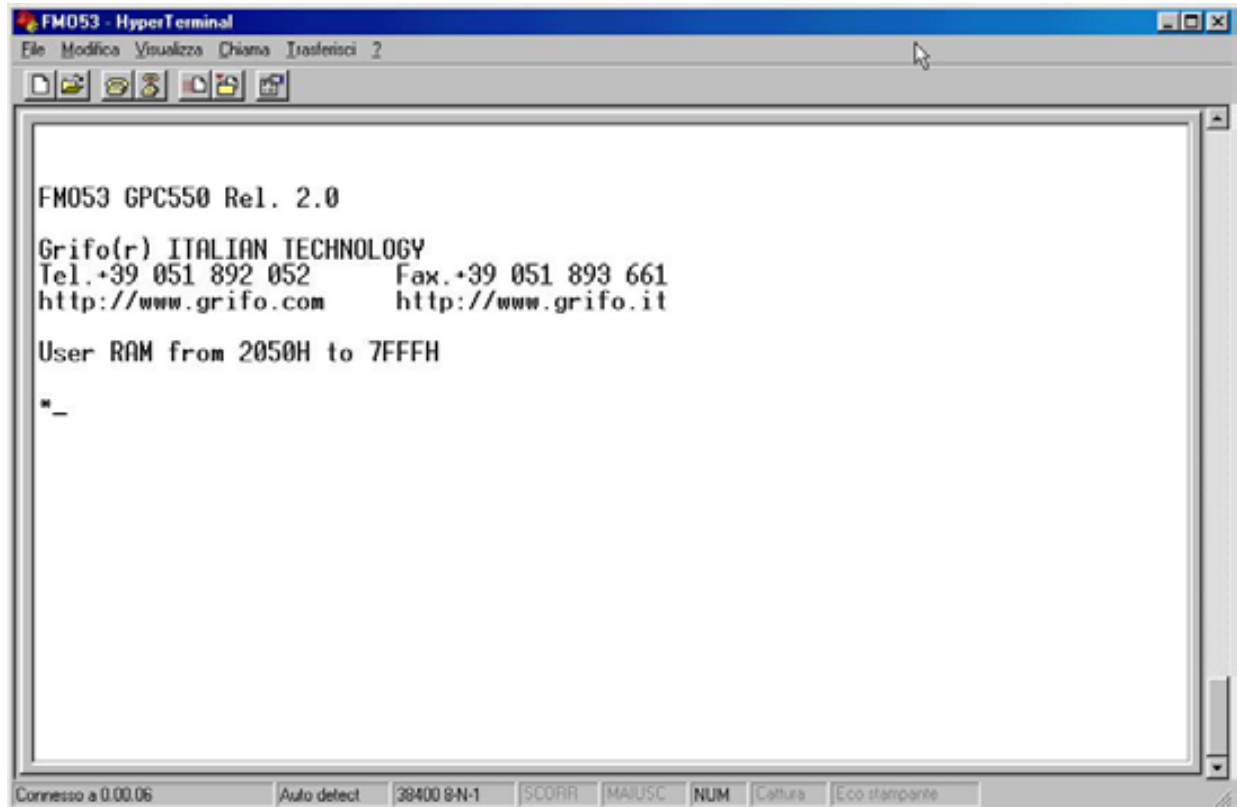


FIGURA 26: MESSAGGIO DI PRESENTAZIONE FM053 SU SERIALE A

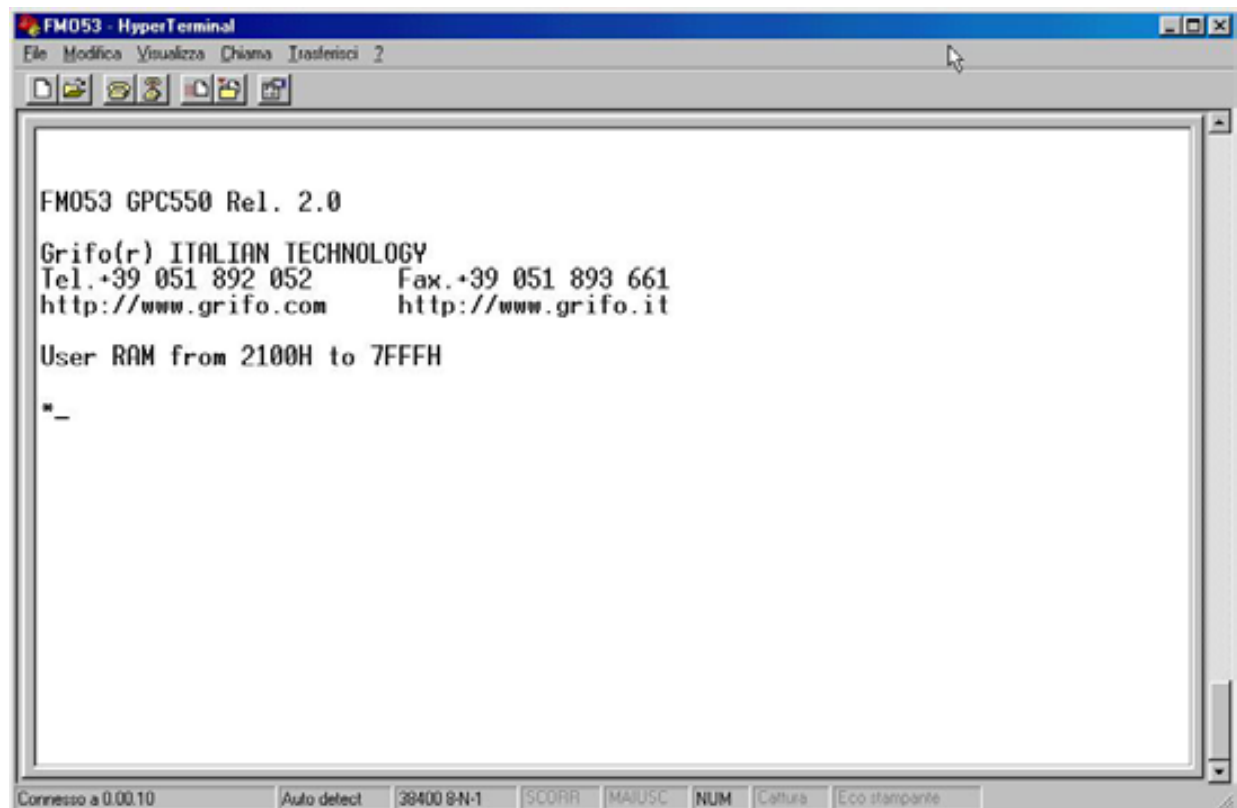
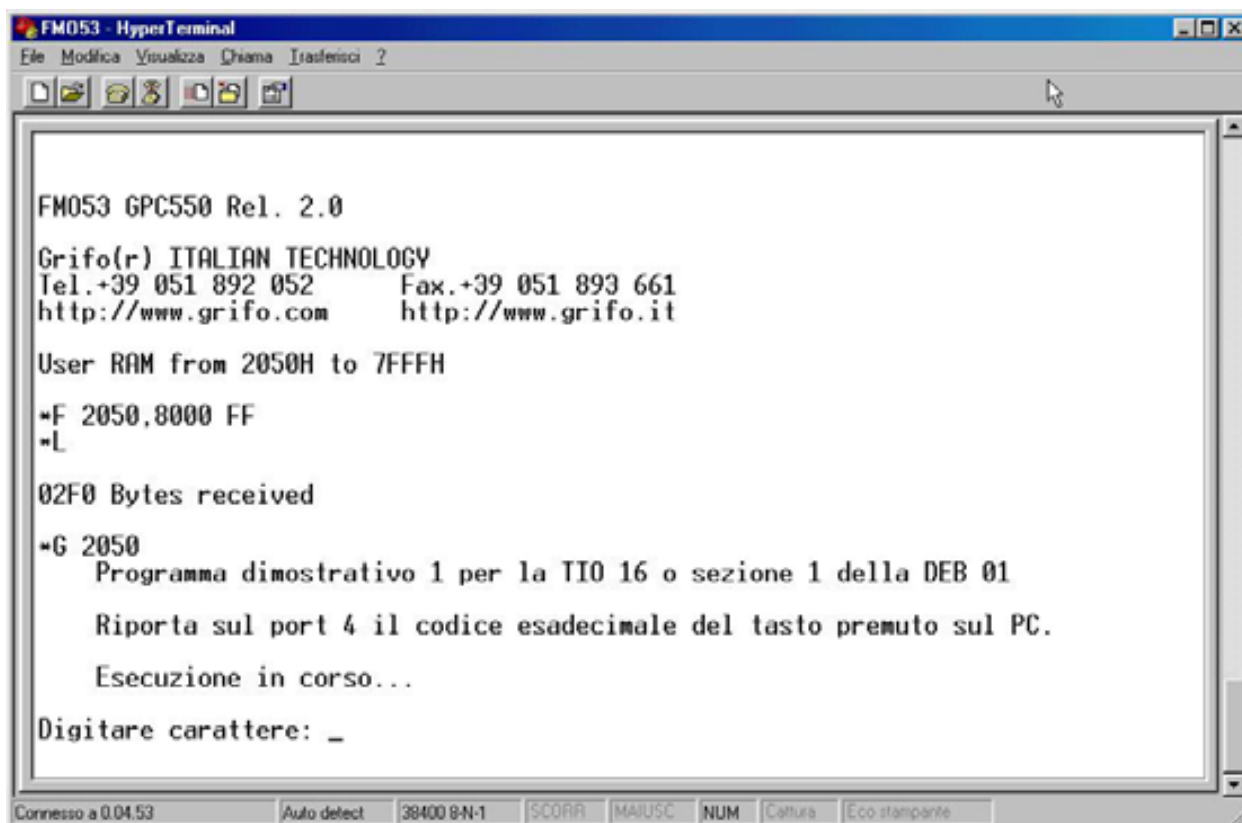


FIGURA 27: MESSAGGIO DI PRESENTAZIONE FM053 SU SERIALE B

- 10) Fornire alcuni comandi e verificarne il funzionamento controllando i risultati rappresentati. Tra questi suggeriamo i comandi *T* (Testa SRAM della scheda), *F* (Riempe memoria dati esterna), *D* (Visualizza memoria dati esterna), *R* (Visualizza registri), ecc.

- 11) Individuare uno dei programmi di esempio della scheda in uso sul CD **grifo**[®] ricevuto, codificato con l'ambiente di sviluppo scelto, e copiarlo in una cartella di lavoro sul PC di sviluppo. Nel caso l'esempio usi anche altri file, provvedere a copiarli tutti.
- 12) Aprire il programma demo descritto al punto 11 con l'ambiente di sviluppo scelto e configurarlo per la modalità DEBUG sulla linea seriale scelta al punto 2. Questa configurazione coincide con il settaggio di entrambi gli indirizzi di **Inizio codice utente** e di **Inizio dati utente** in SRAM, abbondantemente descritti nei paragrafi CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE e DIRETTIVE DEL PROGRAMMA APPLICATIVO.
- 13) Compilare il programma demo aperto, verificando che non si presentino errori e che venga creato il file HEX Intel, con il codice eseguibile dello stesso demo.
- 14) Scaricare il programma HEX Intel ottenuto al punto 13 nella SRAM della scheda di controllo ed eseguirlo, seguendo le indicazioni dei paragrafi CONFIGURAZIONI PER COMUNICAZIONE SERIALE. La foto 28 illustra l'esecuzione di questo passo.



```

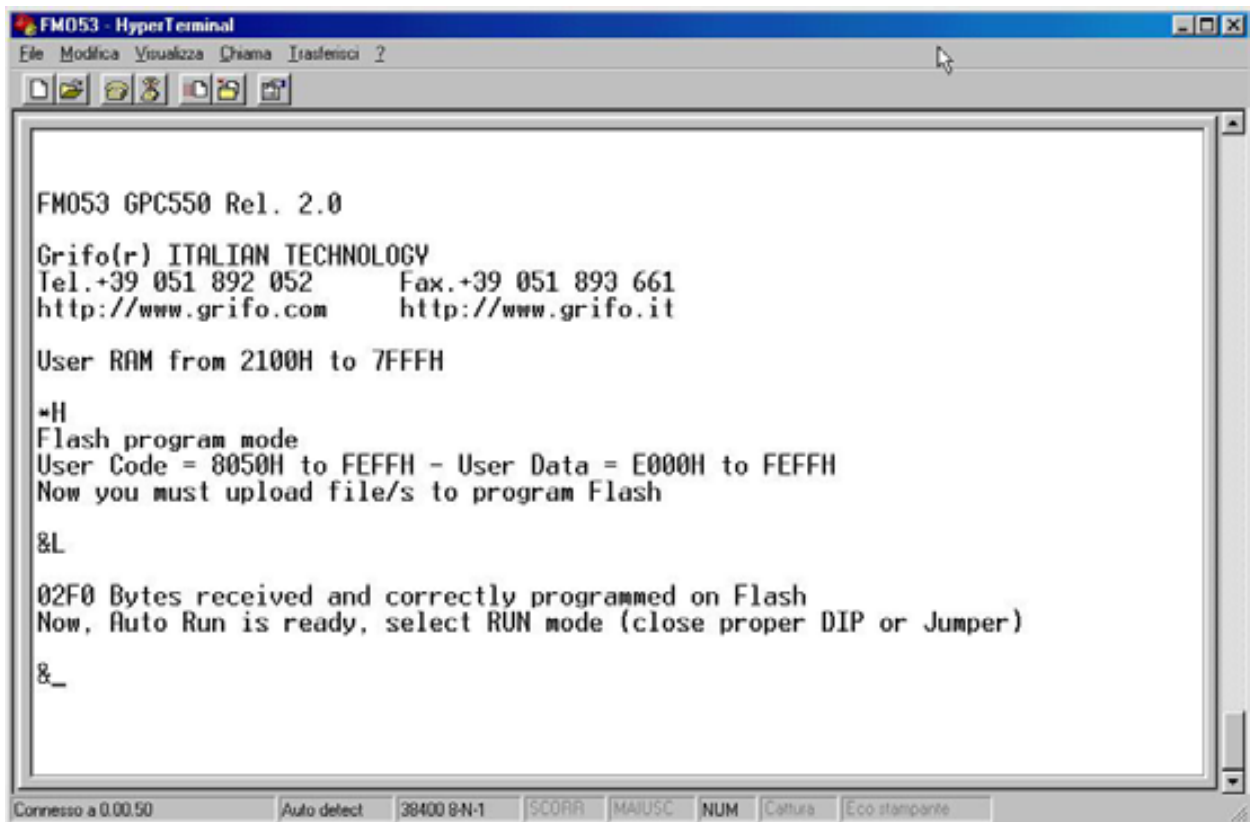
FM053 GPC550 Rel. 2.0
Grifo(r) ITALIAN TECHNOLOGY
Tel.+39 051 892 052      Fax.+39 051 893 661
http://www.grifo.com    http://www.grifo.it

User RAM from 2050H to 7FFFH
+F 2050,8000 FF
+L
02F0 Bytes received
+G 2050
  Programma dimostrativo 1 per la TIO 16 o sezione 1 della DEB 01
  Riporta sul port 4 il codice esadecimale del tasto premuto sul PC.
  Esecuzione in corso...
Digitare carattere: _
    
```

FIGURA 28: SCARICAMENTO ED ESECUZIONE DI UN PROGRAMMA IN SRAM

- 15) Verificare che il programma eseguito funzioni correttamente in ogni sua parte.
- 16) Riconfigurare l'ambiente di sviluppo per la modalità AUTORUN. Anche questa configurazione implica il settaggio dell'indirizzo di **Inizio codice utente** su FLASH e di **Inizio dati utente** su SRAM, sempre abbondantemente descritti nei paragrafi CONFIGURAZIONE PROGRAMMA APPLICATIVO UTENTE e DIRETTIVE DEL PROGRAMMA APPLICATIVO.
- 17) Compilare il programma demo aperto, verificando che non si presentino errori e che venga creato il file HEX Intel con il codice eseguibile dello stesso demo.

18) Scaricare il programma HEX Intel ottenuto al punto 17 nella FLASH della scheda di controllo, seguendo le indicazioni del paragrafo CODICE SU FLASH. La foto 29 illustra l'esecuzione di questo passo.



```
FMO53 GPC550 Rel. 2.0
Grifo(r) ITALIAN TECHNOLOGY
Tel.+39 051 892 052 Fax.+39 051 893 661
http://www.grifo.com http://www.grifo.it
User RAM from 2100H to 7FFFH
=H
Flash program mode
User Code = 8050H to FEFFH - User Data = E000H to FEFFH
Now you must upload file/s to program Flash
&L
02F0 Bytes received and correctly programmed on Flash
Now, Auto Run is ready, select RUN mode (close proper DIP or Jumper)
&_
Connesso a 0.00.50 Auto detect 38400 8-N-1 SCORR MAIUSC NUM Cattura Eco stampante
```

FIGURA 29: SCARICAMENTO E PROGRAMMAZIONE DI UN PROGRAMMA IN FLASH

- 19) Verificare che lo scaricamento e la programmazione siano avvenute correttamente in ogni loro parte. A questo punto in FLASH è stato programmato il programma demo ed i settaggi che lo fanno partire in AUTORUN.
- 20) Impostare il selettore della scheda di controllo in modo da selezionare la modalità AUTORUN, così come descritto nel paragrafo SELEZIONE MODO OPERATIVO.
- 21) In questa fase resettare o riaccendere la scheda per ottenere l'avvio automatico del programma demo salvato in FLASH, come visualizzato nella figura 30.
- 22) Reimpostare il selettore della scheda di controllo in modo da selezionare la modalità DEBUG, così come descritto nel paragrafo SELEZIONE MODO OPERATIVO.
- 23) Resettare o riaccendere la scheda per ottenere di nuovo la partenza dell'**FMO53**, già descritta al punto 9 e nelle figure 26 e 27.

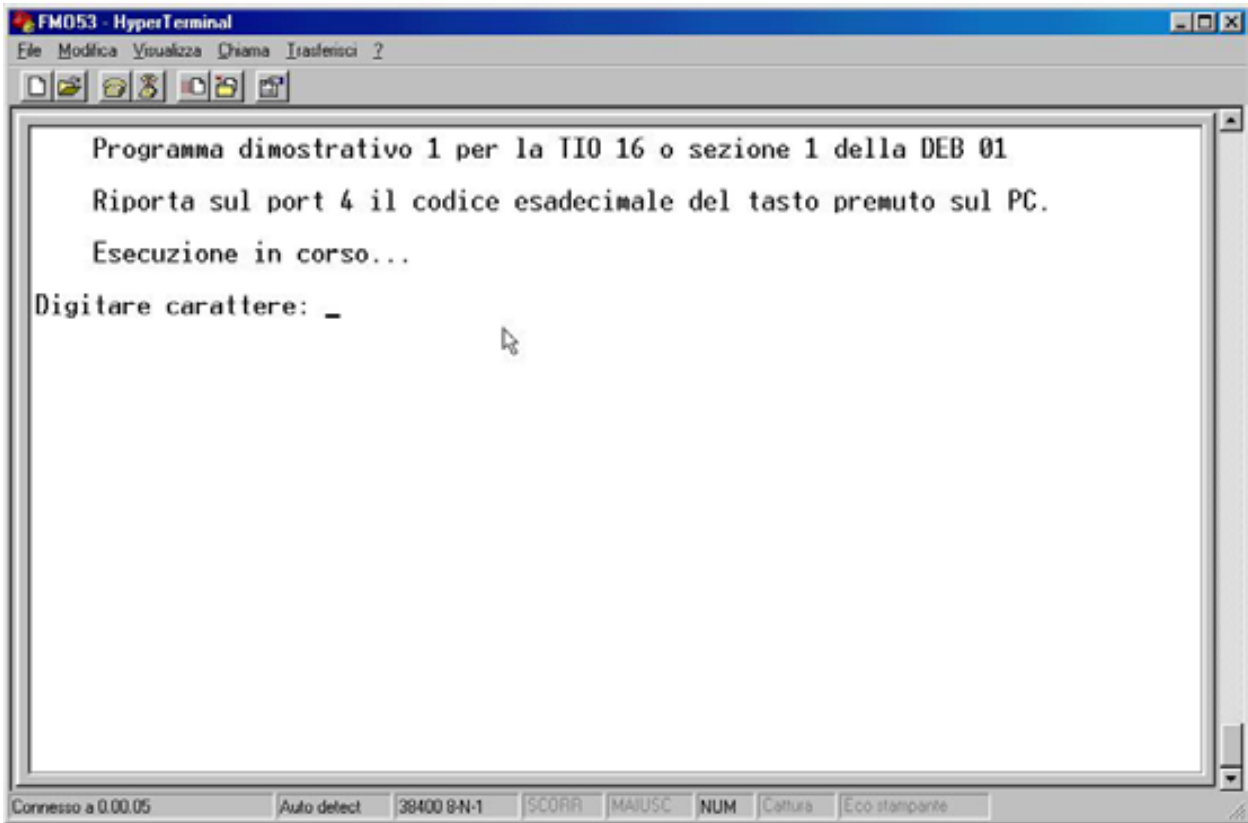


FIGURA 30: PARTENZA AUTOMATICA PROGRAMMA IN AUTORUN

APPENDICE A: INDICE ANALITICO

Simboli

\$baud 28
\$large 28
\$ramsize 28
\$ramstart 28
\$romstart 28
.128K 7, 15
.128KF 7, 15
.32K 7, 14
.32KF 7, 13, 14, 23
.FMO53 13
/PSEN 13
/RD 13
µC/51 10, 34

A

Abilita comandi per FLASH, comando 40
Accessori 9
Altera memoria interna, comando 38
Ambiente di sviluppo 10, 21, 24, 25, 44
Area codice 16, 20
Area dati 16, 20
Assistenza 1
AUTORUN 13, 17, 21, 22, 24, 46

B

BASCOM 8051 10, 25
Baud Rate 19
Baud rate 29
Bit per carattere 19
Breakpoint 38

C

Cancella FLASH, comando 43
Carattere di pronto 37, 40
Carica file, comando 41
Cariche elettrostatiche 1
Cavo seriale 8
Codice su FLASH 22
Codici d'ordine 13
Collegamenti 8
Comandi 37, 45
Come iniziare 44
Comunicazione seriale 8, 19
 µC/51 35
 BASCOM 8051 25
 GET51 29
 HYPERTERMINAL 31
COMx 8, 25, 29, 32

Configurazione programma applicativo 20
Configurazione schede 13, 14, 15
Connettori 9
Contentitore 1
Current loop 19

D

Dati su FLASH 22, 23
DDS MICRO C 51 10
DEBUG 16, 20, 24, 46
Descrizione 13
Dimensioni massime aree di memoria 21
Dip switch 19, 24
Direttive 1, 21
Direttive programma applicativo
 μC/51 35
 BASCOM 8051 27
Diritti 2
Disassembla codice, comando 43

E

Elenco comandi 37
EPROM 7, 12, 13, 18, 21
Esegue a singolo passo, comando 42
Esegue a velocità reale, comando 40
ETHERNET 8
Evoluzione 3
External Ram Start (Hex) 35

F

Fine codice utente 16, 20
Fine dati utente 16, 20
FLASH 7, 12, 16, 17, 18, 22, 40, 47
FMO52 3

G

Garanzia 1
GET51 12, 29, 35
GPC® 7, 14

H

HI TECH C 51 10
HYPERTERMINAL 12, 31, 35, 44

I

I/O di bordo 16
IDE 25
Indirizzi 16, 20, 21
Informazioni generali 4

Ingresso configurazione 18
Inizio codice utente 16, 20, 24, 28, 35, 46
Inizio dati utente 16, 20, 28, 35, 46
Integrazione 25
Interrupt 16, 18, 24
Introduzione 1

J

Jumpers 19, 24
Jumpers memorie 13

L

LADDER WORK 10
Linea seriale 9, 19, 44
Linker 35

M

Mappaggio memorie 13, 16, 20, 21
Marchi registrati 2
Materiale necessario 7
Memorie di massa 8
Messaggi errore 41
Messaggio presentazione 45
Modalità di lavoro 11
Modifica memoria dati esterna, comando 39
Modifica registri, comando 39
Modo operativo 24
Monitor 8
Mouse 8

O

Opzioni 7
Organizzazione memorie 16, 17

P

Parametri 23
Parità 19
Pausa rappresentazioni 37
Personal Computer 7, 44
Procedura risposta interrupt 24
Procedura scrittura dati su FLASH 23
Programma applicativo 22, 24
programma applicativo 20
Programma emulazione terminale 12
Programma utente 20
Programmatore di EPROM 12, 21
Programmazione FLASH 22, 47
Protezioni 1, 22
Protocollo fisico 19, 25, 32
Punto d'arresto 38

R

RAM interna 18, 38, 40
Rappresentazioni 37
Registri 39, 42
Ridirezione 24
Riempe memoria dati esterna, comando 40
Risorse utilizzate 18
Ritardo 24
Rom Start (Hex) 35
RS 232 8, 19
RS 422 19
RS 485 19

S

Scaricamento programmi
 μC/51 35
 BASCOM 8051 27
 GET51 30
 HYPERTERMINAL 33
Scheda di controllo 7, 13
Selettore modalità 18, 24
Setta e visualizza punti d'arresto, comando 38
Sicurezza 1
Sistema operativo 7
Software 9
SRAM 7, 13, 16, 18
Stack 18
Stop Bit 19
SYS51CW 10
SYS51PW 10

T

Testa SRAM della scheda, comando 42
Timer1 18, 42
Tipo terminale 30

U

USB 8

V

Versione 3, 7
Vettori interrupt 16, 24
Visualizza comandi, comando 43
Visualizza memoria dati esterna, comando 39
Visualizza memoria interna, comando 40
Visualizza registri, comando 42

Z

Zoccolo multimemoria 13