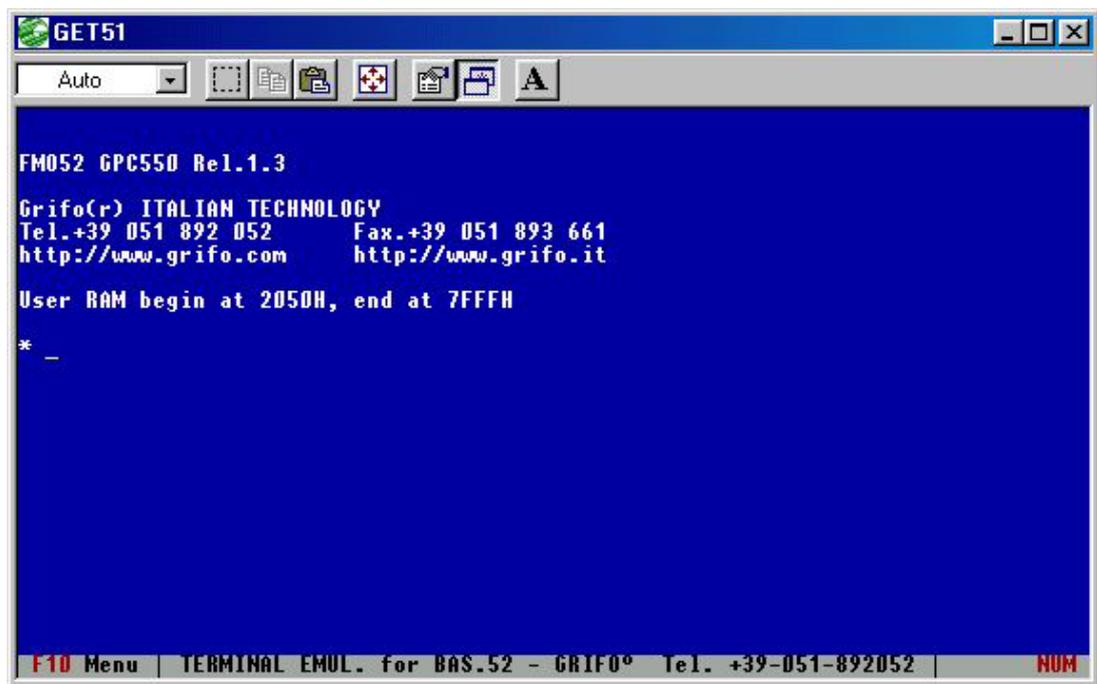


FMO 52

Flash Monitor Debugger per famiglia 51

MANUALE UTENTE



grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it

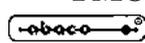


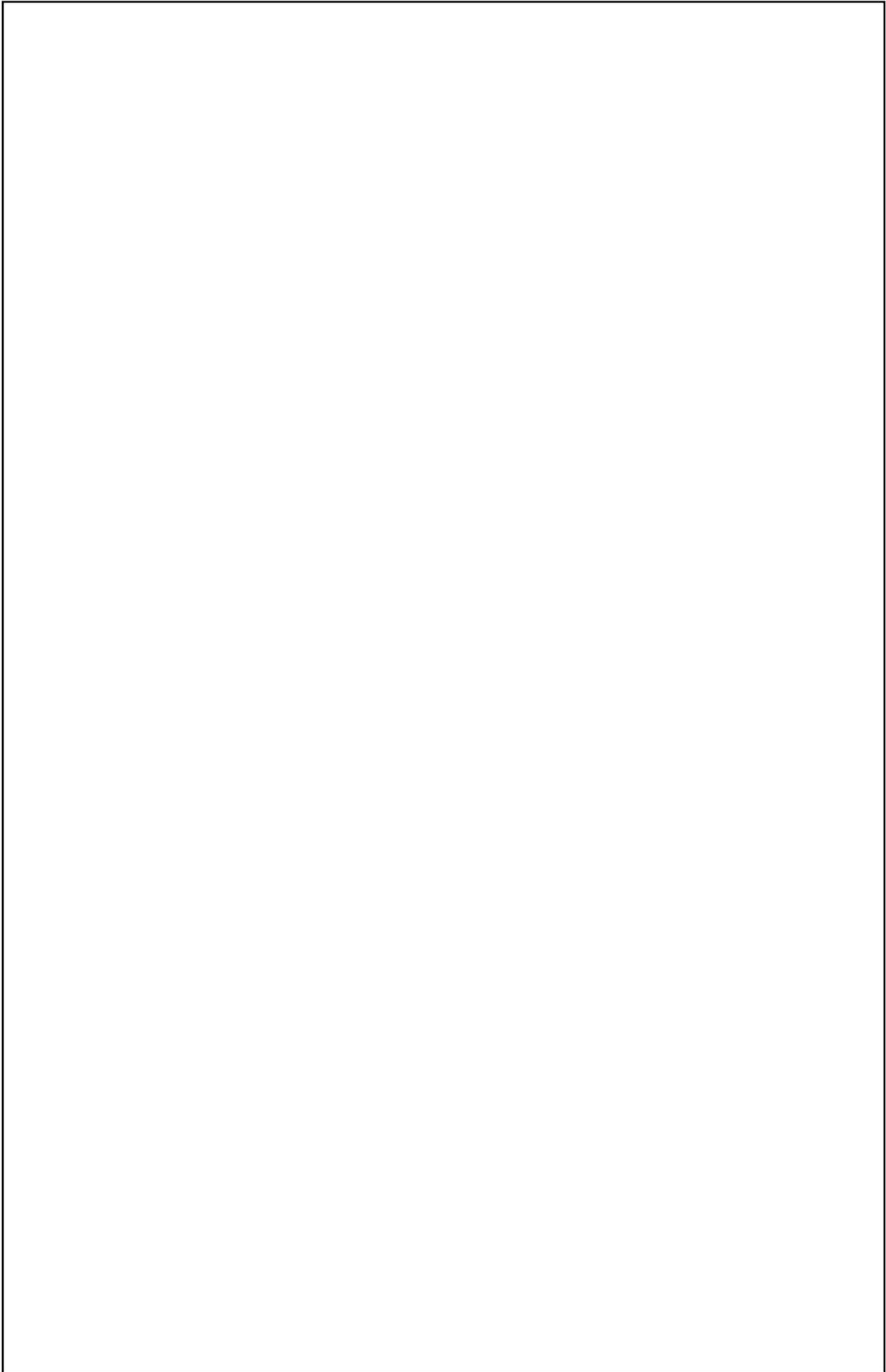
<http://www.grifo.it> <http://www.grifo.com>
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661

FMO 52

Rel. 3.00

Edizione 03 Marzo 2003

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



FMO 52

Flash Monitor Debugger per famiglia 51

MANUALE UTENTE

FMO52 e' un pacchetto software per la famiglia di microprocessori Intel 51 interattivo che comprende un debugger, un monitor in linguaggio macchina che permette il debug del programma utente.

FMO52 richiede l'installazione di una Flash **ATMEL AT29c256** per la sola gestione della programmazione e dell'esecuzione del programma utente e sfrutta pochissime risorse dell'hardware utilizzato.

Include funzioni come

- visualizzazione ed alterazione del contenuto della memoria
- gestione di punti d'arresto
- esecuzione codice a singola istruzione
- esecuzione codice a velocità reale
- un completo disassemblatore
- test della SRAM
- cancellazione della Flash
- programmazione della Flash con il programma utente
- lancio del programma in AUTO RUN

grifo[®]
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it



<http://www.grifo.it>

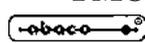
<http://www.grifo.com>

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

FMO 52

Rel. 3.00

Edizione 03 Marzo 2003

, **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:

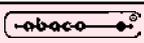


Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

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
INFORMAZIONI PRELIMINARI	2
INSTALLAZIONE	4
UTILIZZO DI FMO 52 CON BASCOM 8051	7
UTILIZZO DI FMO 52 CON GET 51	10
COMPILAZIONE CON BASCOM 8051 PER FMO 52	14
GENERAZIONE DEI PROGRAMMI DI PROVA	14
GENERAZIONE DEI PROGRAMMI DEFINITIVI	16
UTILIZZO DELLA FLASH DA PROGRAMMA UTENTE	18
INTERRUPT.....	19
COMANDI DI FMO 52	20
RAPPRESENTAZIONE	20
ELENCO.....	20
APPENDICE A: INDICE ANALITICO	A-1

INDICE DELLE FIGURE

FIGURA 1: LA SCHERMATA INIZIALE	3
FIGURA 2: MAPPA DELLA MEMORIA PER FMO 52.....	4
FIGURA 3: CONFIGURAZIONE SCHEDE GPC® PER FMO 52	5
FIGURA 4: CONFIGURAZIONE DELLA SERIALE DEL BASCOM 8051	7
FIGURA 5: PARAMETRI DI COMUNICAZIONE CON FMO 52	8
FIGURA 6: SCARICAMENTO DEL FILE INTEL HEX CON BASCOM 8051	9
FIGURA 7: CONFIGURAZIONE DELLA SERIALE DEL GET 51	10
FIGURA 8: PARAMETRI DI COMUNICAZIONE CON FMO 52	11
FIGURA 9: SCARICAMENTO DEL FILE INTEL HEX CON GET 51	11
FIGURA 10: ELENCO DEI COMANDI DI FMO 52	13
FIGURA 11: INGRESSO NELLA MODALITÀ DI PROGRAMMAZIONE DELLA FLASH	16
FIGURA 12: PROGRAMMAZIONE DELLA FLASH	17

INTRODUZIONE

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

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

Al fine di non incontrare problemi nell'uso di tali dispositivi, é conveniente che l'utente - **PRIMA DI COMINCIARE AD OPERARE** - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

Il programma qui descritto è coperto da diritto d'autore, tutti i diritti sono riservati. Nè il programma nè alcuna sua parte possono essere analizzati, disassemblati o modificati in alcun modo, con qualunque mezzo, per qualunque scopo.

Questo documento è coperto da diritto d'autore, tutti i diritti sono riservati. Questo documento non può essere copiato, riprodotto o tradotto in alcun modo o mediante alcun mezzo, nè per intero nè in parte, senza il permesso scritto della **grifo®**.

grifo® non si assume alcuna responsabilità per l'uso errato di questo manuale.

grifo® si riserva il diritto di apportare cambiamenti o miglioramenti ai prodotti descritti in questo manuale in qualunque momento senza darne notizia.

Questo manuale contiene nomi di aziende, software, prodotti, ecc. che sono marchi registrati dai rispettivi proprietari. **grifo®** rispetta tale diritto.

INFORMAZIONI PRELIMINARI

FMO52 e' un pacchetto software per la famiglia di microprocessori Intel 51 interattivo che comprende un debugger, un monitor in linguaggio macchina che permette il debug del programma utente nell'area di RAM da 2050H fino 7FFFH, dove e' indispensabile sezionarla in area codice e area dati.

Se il codice viene compilato per l'esecuzione in Flash da 8050H fino DFFFH, l'area di RAM risulta essere tutta disponibile per i dati da 20C6H fino 7FFFH, questo accade nella modalita' di AUTO RUN.

FMO52 richiede l'installazione di una Flash **ATMEL AT29c256** per la sola gestione della programmazione e dell'esecuzione del programma utente e sfrutta pochissime risorse dell'hardware utilizzato.

In effetti l'**FMO52** usa la linea seriale per la comunicazione nei confronti del P.C. di sviluppo, 80 bytes di SRAM esterna (da 2000H a 204FH) per le sue variabili, 6631 bytes di ROM per il suo codice ed il **TIMER1** per la generazione del **baud rate** di comunicazione e per l'**esecuzione passo passo**. Al fine di poter caricare un programma applicativo dal P.C. di sviluppo e di poter settare dei punti d'arresto l'**FMO52** deve essere in grado di poter scrivere nell'area codice.

Visto che l'architettura dell'8051 non consente queste operazioni quando si usa un'area codice separata da quella dati, e' necessario che l'area SRAM di scaricamento del programma sia indirizzata sia come dati che come codice (**/PSEN e /RD sommati**).

FMO 52 è una opzione ordinabile esplicitamente delle schede del carteggio **grifo®** che lo supportano (si veda l'elenco dei nomi nella tabella di figura 2).

Per ordinarlo si deve estendere il nome della scheda con il codice:

.FMO52

mentre per ordinare anche la FLASH l'esternione diventa:

.FMO52.32KF

Include funzioni come:

- visualizzazione ed alterazione del contenuto della memoria
- gestione di punti d'arresto
- esecuzione codice a singola istruzione
- esecuzione codice a velocità reale
- un completo disassemblatore
- test della SRAM
- cancellazione della Flash
- programmazione della Flash con il programma utente
- lancio del programma in AUTO RUN DIP-SWITCH o Jumper all'accensione

Il presente manuale è riferito all'**FMO 52 versione 1.2 o superiori** per tutte le **GPC®** supportate.



FIGURA 1: LA SCHERMATA INIZIALE

INSTALLAZIONE

FMO 52 è un programma downloader, monitor e debugger per microcontrollori della famiglia 51. Per il suo funzionamento richiede che la memoria dell'hardware su cui gira si organizza nel seguente modo:



FIGURA 2: MAPPA DELLA MEMORIA PER FMO 52

Le schede **grifo**® della famiglia 51 supportano, tra gli altri, anche questo mappaggio della memoria. Quasi tutte sono adatte (e quindi dotate) di una propria versione di **FMO 52**.

La tabella di figura 3 elenca tutte le schede dotate di **FMO 52** e le istruzioni per configurare il mappaggio di memoria, il tipo di memorie, l'installazione delle memorie e la gestione delle modalità di AUTO RUN e DEBUG.

Per informazione sull'indirizzo di **fine della FLASH** si prega di consultare il manuale della scheda.

FMO 52 normalmente viene fornito come opzione ordinata esplicitamente, già installata e configurata sulla scheda **GPC**® che avete chiesto.

In tal caso sono presenti:

- Una EPROM con etichetta “**FMO 52** Rel. x.xx **GPC**® yyyy” dove x.xx indica la versione dell'**FMO 52** nella EPROM e yyyy indica per quale scheda **GPC**® è adatto
- Una SRAM da 32KBytes
- Opzionalmente, una FLASH Atmel 29c256 da 256 KBytes

In caso aveste ordinato **FMO 52** a parte o in un secondo tempo, la tabella di figura 3 spiega come installare la EPROM e configurare la **GPC**® per far eseguire **FMO 52**.

Per identificare gli zoccoli ed i jumpers si prega di consultare il manuale della scheda. In dettaglio:

- **Scheda** indica per quale scheda valgono le istruzioni
- **Mappaggio** indica il nome del mappaggio di memoria come riportato del manuale della scheda
- **Selezione mappaggio** indica quali jumper e/o dip switch muovere per ottenere il mappaggio
- **EPROM** indica in quale zoccolo inserire la EPROM con etichetta **FMO 52**
- **SRAM** indica in quale zoccolo inserire la SRAM (se assente)
- **FLASH** indica in quale zoccolo inserire la FLASH opzionale per il programma utente definitivo
- **AUTO RUN** indica l'impostazione per far partire all'avvio il programma utente nella FLASH
- **Jumpers Memorie** indica quali jumpers muovere per configurare le memorie

Scheda	Mappaggio Memorie	Selezione Mappaggio	EPROM	RAM	FLASH	AUTO RUN	Jumpers Memorie
GPC® 323/D Rel. 110197	MODO 1	DIP 5 ON DIP 6 ON	IC 5	IC 4	IC 3	DIP 8 ON	J4 e J5 in 1-2 J2 in 1-2 J3 pin 2 con J2 pin 3
GPC® 323/D Rel. 250601	MODO 3	DIP 5 ON DIP 6 ON	IC 5	IC 4	IC 3	DIP 8 ON	J2 in 1-2 e 3-4
GPC® 324/D Rel. 100997	MODO 3	J2 chiuso J3 in 2-3	IC 8	saldata	IC 5	J1 chiuso	J4 in 1-2 J5 pin 2 con J4 pin 3
GPC® 324/D Rel. 110400	MODO 3	J2 chiuso J3 in 2-3	IC 8	saldata	IC 5	J1 chiuso	J4 in 1-2 e 3-4
GPC® 552	MODO 1	J1 in 3 J2 in 3 J14 in 1,2,3 ASM	IC 15	IC 13	IC 12	DIP 8 ON	J18 in 1-2 J17 pin 2 con J18 pin 3
GPC® 553	MODO 3	DIP 5 ON DIP 6 ON	IC 1	IC 2	IC 3	DIP 8 ON	J4 e J5 in 1-2 J2 in 1-2 J3 pin 2 con J2 pin 3
GPC® 554	MODO 3	J1 chiuso J6 in 2-3	IC 4	saldata	IC 6	J2 chiuso	J3 in 1-2 J4 pin 2 con J3 pin 3
GPC® R/T63	MODO 3	J2 chiuso J3 in 2-3	IC 7	saldata	IC 10	J1 chiuso	J4 in 1-2 J3 pin 2 con J4 pin 3
GPC® 550	MODO 3	J4 in 2-3 J5 chiuso	U1	saldata	U3	DIP 8 ON	J1 in 1-2 e 3-4

FIGURA 3: CONFIGURAZIONE SCHEDE GPC® PER FMO 52

Una volta installate le memorie e configurati i jumpers e/o i dip switch, collegare la porta seriale principale della scheda **GPC®** alla porta seriale del PC o dell'emulatore di terminale.

Per sapere qual è la porta seriale principale e quale cavo utilizzare si prega di riferirsi al manuale della scheda **GPC®**.

FMO 52 lavora bene insieme a compilatori con IDE che incorporano anche un emulatore terminale e prevedono esplicitamente modalità di scaricamento del file compilato in formato Intel Hex o S Motorola.

Un esempio è il **BASCOM 8051**, compilatore basic economico, efficiente e facile da usare.

Un prodotto **grifo®** inoltre, l'emulatore terminale **GET 51**, nasce con una modalità specifica per gestire la comunicazione con **FMO 52**.

Le istruzioni specifiche per configurare i prodotti sopra citati si trovano negli appositi paragrafi del presente manuale, in generale le istruzioni per configurare un qualunque emulatore di terminale sono le seguenti:

- | | |
|---------------|---------|
| - Baud Rate | 19200 |
| - Bit di dati | 8 |
| - Parità | Nessuna |
| - Bit di stop | 1 |

Una volta avviato e configurato l'emulatore di terminale, alimentate o resettate la scheda **GPC®**. All'accensione o dopo un reset, **FMO 52** verifica se è impostata la condizione di **AUTO RUN** o quella di **DEBUG**.

- nel primo caso viene immediatamente messo in esecuzione, se presente, il programma utente scritto nella Flash, altrimenti partirà **FMO 52** regolarmente
- nel secondo caso verrà subito eseguito **FMO 52**, normalmente serve per poter scaricare un programma in SRAM e debuggarlo.

Nella figura 1 viene mostrata la schermata iniziale che dovete riuscire a vedere se la **GPC®** è collegata correttamente all'emulatore di terminale ed è stata configurata correttamente.

Se non la doveste vedere, controllate tutte le impostazioni della tabella di figura 3.

Dal prompt '*' potete scaricare il vostro programma da un file in formato **Intel HEX** (con il comando l) oppure iniziare una sessione di debugging o di dumping dei dati già presenti avvalendovi dei potenti comandi che verranno illustrati in seguito.

UTILIZZO DI FMO 52 CON BASCOM 8051

BASCOM 8051 è un **cross-compiler** basic economico, efficiente e facile da usare, provvisto di un **completo IDE**, che gira sotto qualunque versione di Windows e genera codice eseguibile per microcontrollori della **famiglia 51**.

Una delle funzionalità dell'IDE consiste in un emulatore terminale che gira in una finestra Windows ed è configurabile sia nei parametri classici della porta seriale, sia nel comportamento.

Infatti è possibile attivare, mediante un comodo menu, l'invio automatico di una sequenza di stringhe configurabile dell'utente e del risultato dell'ultima compilazione.

In questo modo **FMO 52** può ricevere dal **BASCOM 8051** il comando di scaricamento di un file Intel Hex, il file stesso, ovvero l'ultimo programma compilato ed il comando di fine scaricamento semplicemente attivando la voce di un menu.

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

- Tramite il menu Options | Communications, configurare i parametri della porta seriale come da figura 4, mettendo nella voce COM port il nome della porta seriale del PC alla quale avete collegato la seriale principale della scheda **GPC®**, e premere OK.

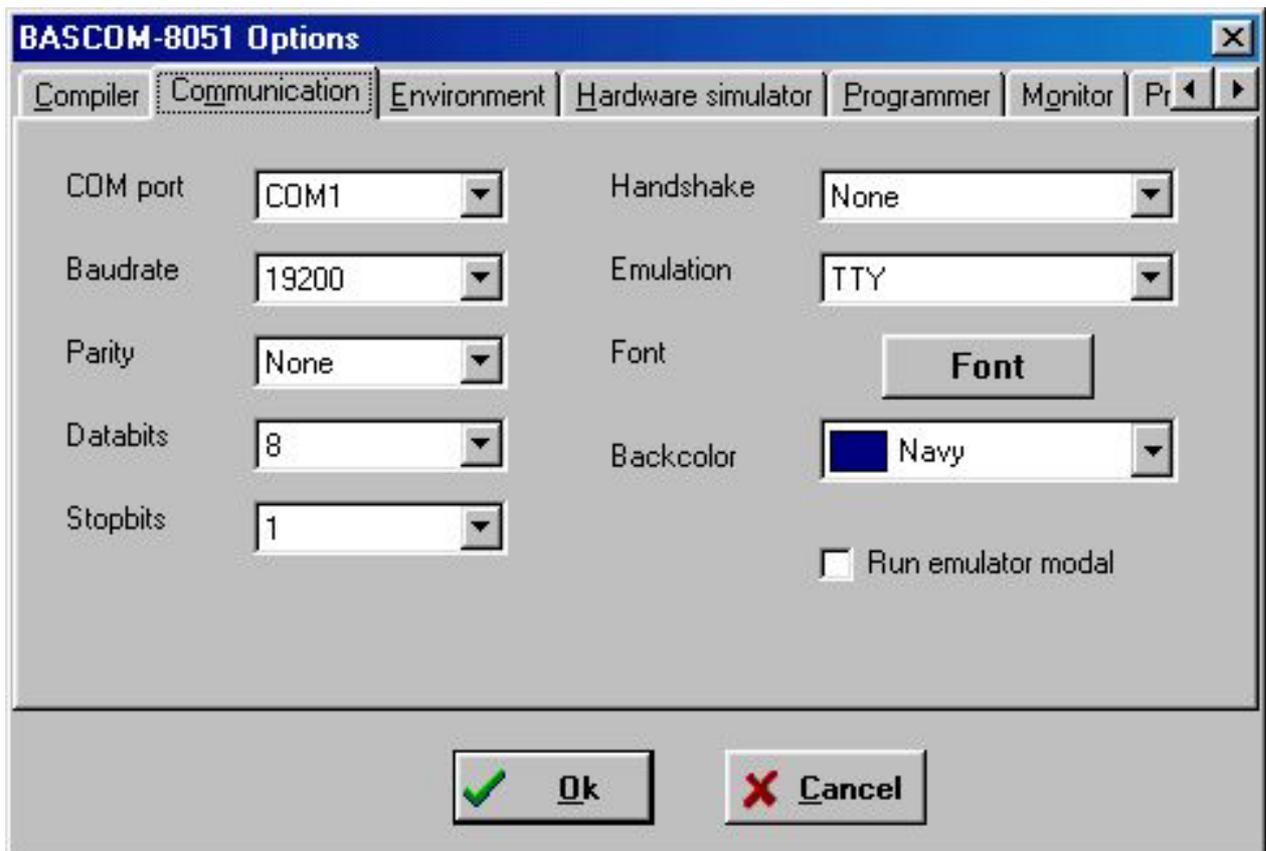


FIGURA 4: CONFIGURAZIONE DELLA SERIALE DEL BASCOM 8051

- Tramite il menu Options | Compiler | Monitor, configurare i parametri della comunicazione con **FMO 52** come da figura 5, e premere OK.

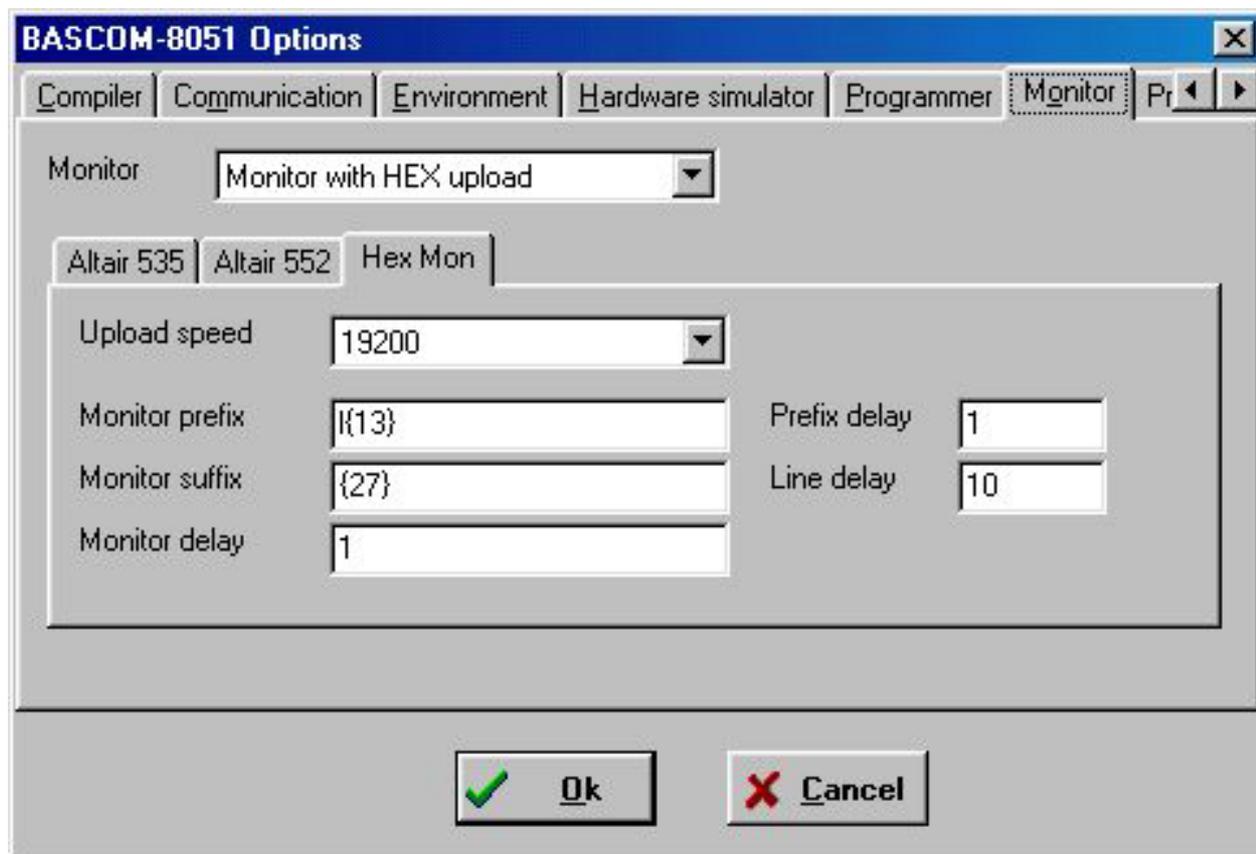


FIGURA 5: PARAMETRI DI COMUNICAZIONE CON FMO 52

- Scrivere il programma sorgente, o caricarlo da file, e compilarlo. Per maggiori informazioni su come compilare un programma **BASCOM 8051** adatto all'**FMO 52**, si prega di riferirsi allo specifico paragrafo di questo manuale. Ogni scheda **GPC®** della famiglia 51 è fornita con un programma demo completa già predisposto per essere compilato con **BASCOM 8051** pensato per agevolare lo sviluppo di applicativi.
- 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.
- Alimentare o resettare la scheda **GPC®**. Dovete vedere la schermata iniziale di figura 1 con il prompt '*'. Ripetete i primi due punti di questa lista se avete dei problemi.
- Iniziate lo scaricamento del file Intel Hex con la voce File | Upload del menu (vedere figura 6).
- Dopo lo scaricamento, l'indicazione esadecimale che appare è un conteggio del numero di bytes scaricati, che coincide con la lunghezza del codice generato arrotondato per eccesso ad un multiplo di 16 (10 esadecimale). Il programma viene eseguito digitando il comando G 2050. Si veda l'apposito paragrafo sui comandi per ulteriori informazioni.

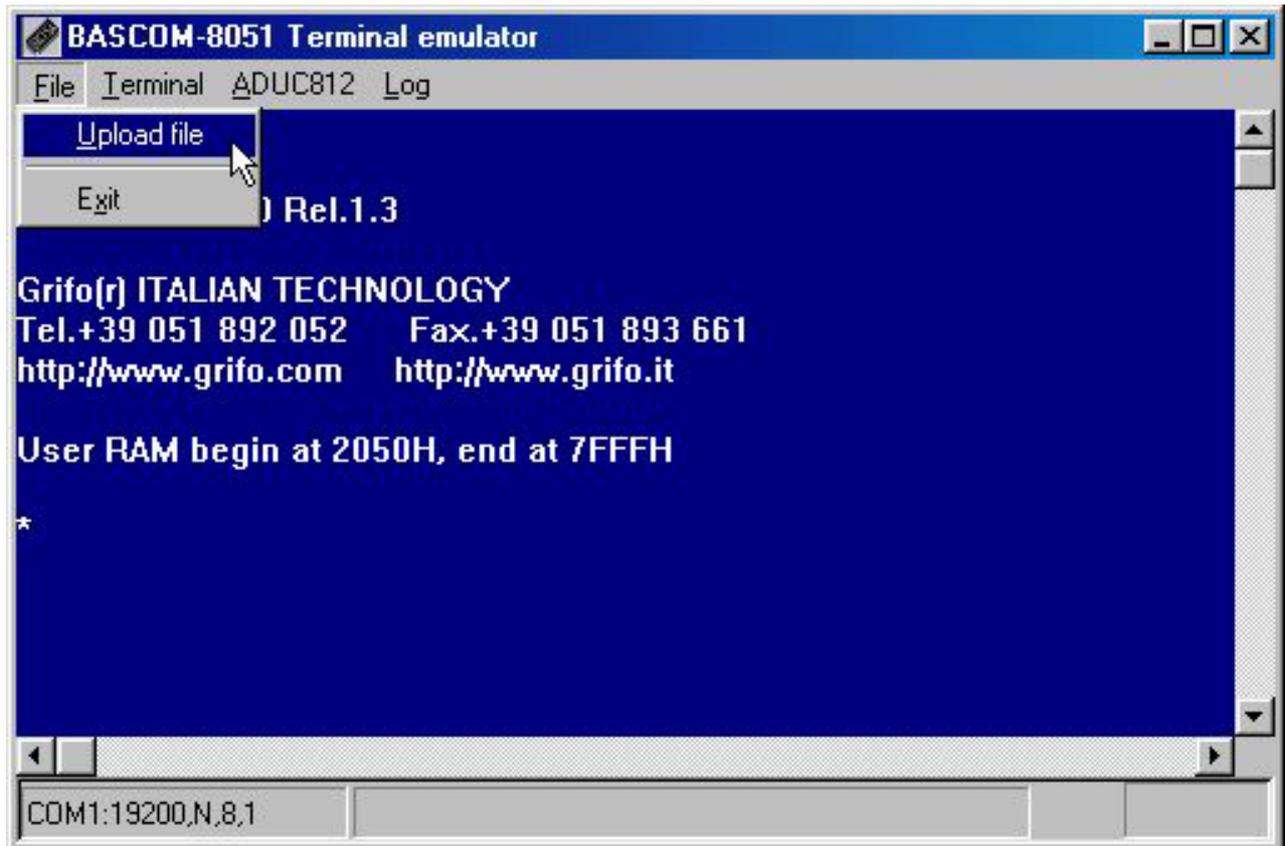


FIGURA 6: SCARICAMENTO DEL FILE INTEL HEX CON BASCOM 8051

UTILIZZO DI FMO 52 CON GET 51

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 l'**FMO 52**.

In questo manuale verrà tralasciato tutto quanto non riguarda direttamente la comunicazione con **FMO 52**, se desiderate ulteriori informazioni potete consultare il manuale on-line del **GET 51** presente sul CD **grifo**® o sul sito Internet www.grifo.it.

Una delle funzionalità del **GET 51** consiste in un emulatore terminale configurabile sia nei parametri classici della porta seriale, sia nel comportamento.

Infatti è possibile attivare, mediante un comodo menu, l'invio automatico di un file Intel Hex verso **FMO 52**.

In questo modo **FMO 52** può ricevere dal **GET 51** il comando di scaricamento di un file Intel Hex, il file stesso ed il comando di fine scaricamento semplicemente attivando la voce di un menu.

Per ottenere questo bisogna configurare **GET 51** nel seguente modo:

- Tramite il menu Options | Serial Port, configurare i parametri della porta seriale come da figura 7, mettendo nella voce COM port il nome della porta seriale del PC alla quale avete collegato la seriale principale della scheda **GPC**®, e premere OK.

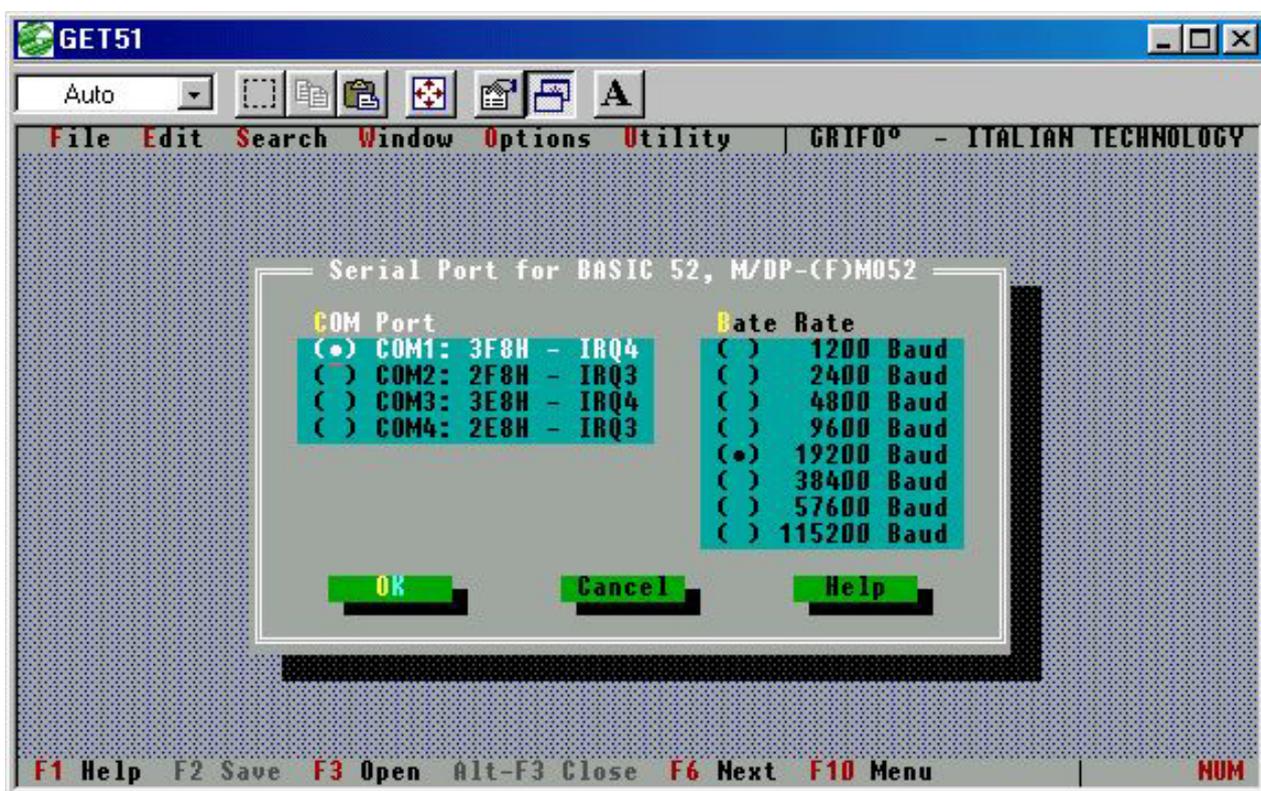


FIGURA 7: CONFIGURAZIONE DELLA SERIALE DEL GET 51

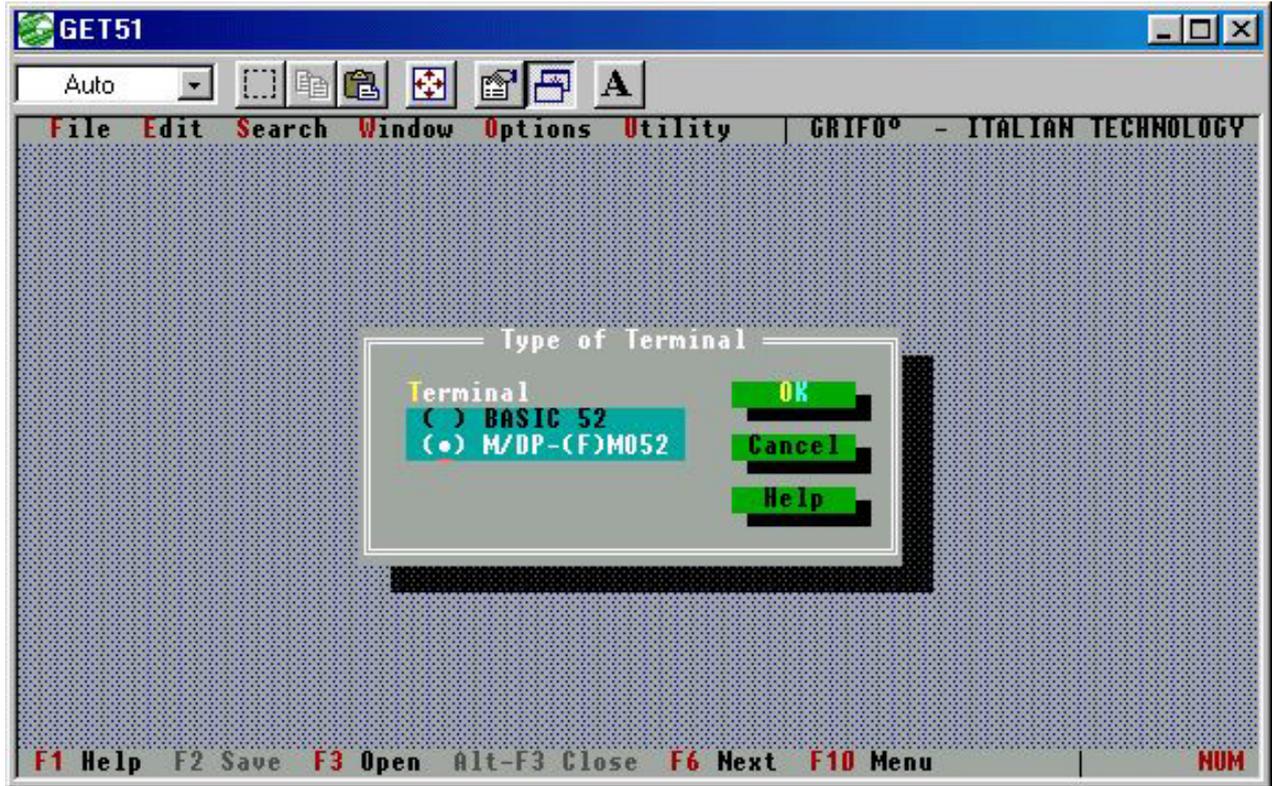


FIGURA 8: PARAMETRI DI COMUNICAZIONE CON FMO 52

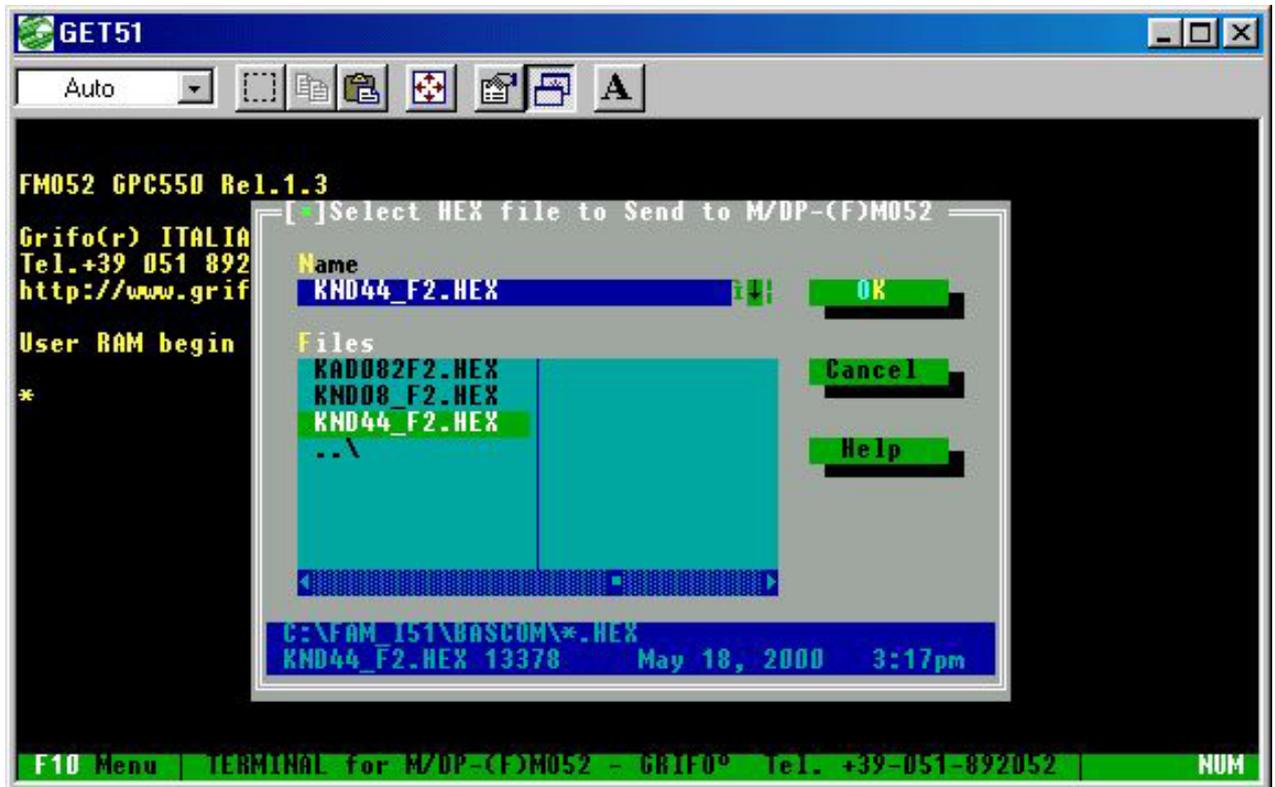


FIGURA 9: SCARICAMENTO DEL FILE INTEL HEX CON GET 51

- Tramite il menu Options | Set Terminal, configurare **GET 51** per la comunicazione con **FMO 52** come da figura 8, e premere OK.

- Entrare nell'emulatore di terminale tramite il menu Options | Terminal oppure premendo il tasto F5.

- Alimentare o resettare la scheda **GPC®**. Dovete vedere la schermata iniziale di figura 1 con il prompt '*'. Ripetete i primi due punti di questa lista se avete dei problemi.

- Per selezionare il file da scaricare, premere il tasto F8 (vedere figura 9). La finestra che appare permette di cercare il file desiderato attraverso tutti i dispositivi di memoria di massa disponibili e di selezionarlo premendo Invio, una volta selezionato premendo di nuovo Invio inizierà subito lo scaricamento.

- Dopo lo scaricamento, l'indicazione esadecimale che appare è un conteggio del numero di bytes scaricati, che coincide con la lunghezza del codice generato arrotondato per eccesso ad un multiplo di 16 (10 esadecimale). Il programma viene eseguito digitando il comando G 2050. Si veda l'apposito paragrafo sui comandi per ulteriori informazioni.

```
BASCOM-8051 Terminal emulator
File Terminal

User RAM begin at 2050H, end at 7FFFH

*?
FM052 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 (download)
R                   - dump Registers
S                   - Single-Step next instruction
T                   - Test RAM
U <aaaa>,[aaaa]      - Un-assemble

***** Flash commands *****
H                   - Debug mode/Flash program mode
W                   - Copy all the RAM from 2050H into Flash from 8050H
X                   - Erase Flash
*
```

COM3:19200,N,8,1

FIGURA 10: ELENCO DEI COMANDI DI FMO 52

COMPILAZIONE CON BASCOM 8051 PER FMO 52

Quando viene eseguita una compilazione ad un sorgente **BASCOM 8051**, viene generato un file Intel Hex che contiene il codice macchina del microcontrollore utilizzato.

Questo file va inserito in una EPROM o un microcontrollore con una ROM interna, ma quando il programma non è definitivo ma ancora in fase di messa a punto, è necessario ripetere la programmazione diverse volte e questo comporta una perdita di tempo notevole.

Per semplificare la fase di sviluppo di un programma e' sufficiente installare il monitor debugger **FMO 52** a bordo della **GPC®** seguendo le istruzioni del capitolo "INSTALLAZIONE" (se non è già installato) ed avvalersi della possibilità di scaricare il programma nella SRAM di bordo della **GPC®**.

In questo modo è possibile provare il programma un numero praticamente illimitato di volte direttamente sulla struttura che verrà utilizzata nell'applicativo definitivo, e senza dover mai riprogrammare e cancellare EPROM.

GENERAZIONE DEI PROGRAMMI DI PROVA

Il compilatore **BASCOM 8051** è in grado di generare binari eseguibili **senza vincoli fissi** sulla frequenza di **clock**, struttura di **memoria** e **CPU** adottata nella scheda su cui girerà il programma. Per modificare ognuno di questi parametri è possibile agire sui menu dell'IDE o, preferibilmente, fornire direttive di compilazione direttamente nel sorgente.

Tali direttive si trovano all'inizio del sorgente e svolgono sia il compito di informare il compilatore su quale hardware dovrà far girare il programma, sia di documentare tali informazioni direttamente nel sorgente, rendendone immediato il riconoscimento da parte del programmatore.

In particolare:

- La direttiva \$regfile specifica l'SFR della particolare CPU che si vuole usare
- La direttiva \$crystal indica la frequenza del quarzo utilizzato **in Hertz**
- La direttiva \$baud indica il baud rate della comunicazione seriale del **programma** (da non confondere con il baud rate proprio dell'**FMO 52**, che è fisso a 19200)
- La direttiva \$romstart indica l'indirizzo del primo byte di codice del programma
- La direttiva \$ramstart indica l'indirizzo del primo byte SRAM esterna disponibile
- La direttiva \$ramsize la dimensione della SRAM esterna

Ci sono altre direttive non direttamente utili per l'**FMO 52**, se desiderate ulteriori informazione potete consultare il manuale on-line del **BASCOM 8051**.

Come detto nel primo capitolo del manuale, **FMO 52** richiede una particolare struttura della memoria, schematizzata in figura 2 (e ottenibile sulle schede **GPC®** con le impostazioni di figura 3), in particolare **cambiano i valori** delle direttive **\$romstart**, **\$ramstart** e **\$ramsize** per il **programma di prova** o per l'**eseguibile definitivo**.

Per il programma di prova i valori da assegnare sono:

- \$romstart = &H2050
- \$ramstart = &H600
- \$ramsize = &H1FFF

Il valore di \$romstart coincide con la prima locazione di SRAM libera per ospitare il codice di prova, essendo le locazioni da 2000H a 204FH occupate dai vettori di interrupt.

Il valore di \$ramstart viene assegnato provvisoriamente, in quanto nello spazio da 2000H a 7FFFH devono convivere **il codice** del programma di prova e **le variabili** in SRAM esterna.

Il valore di \$ramsize è pari alla SRAM non occupata dal codice del programma di prova, anche tale valore è provvisorio.

La direttiva \$regfile dice al compilatore quale deve essere il file di struttura dell'SFR da usare per la prossima compilazione.

Tali file sono di testo, leggibili e scrivibili dal programmatore, e istruiscono il **BASCOM 8051** su quali ulteriori SFR ha il microcontrollore che si vuole usare rispetto all'architettura standard 8051.

In sostanza, ognuno di questi file caratterizza un diverso microcontrollore.

Per ulteriori informazioni si faccia riferimento alla documentazione on-line del **BASCOM 8051**.

L'attuale corrispondenza tra scheda **grifo® GPC®** e tali files è la seguente:

Scheda	File
GPC® R/T63	8052.DAT
GPC® 550	80552m.DAT
GPC® 552	80552m.DAT
GPC® 553	80552m.DAT
GPC® 554	80552m.DAT
GPC® 323/D	87C520.DAT
GPC® 324/D	87C520.DAT

A titolo di esempio, si riporta l'inizio di un tipico sorgente **BASCOM 8051** per **GPC®**:

```

$regfile = "80552m.dat"
$crystal = 22118400 ' frequenza di clock del microcontrollore
$baud = 19200      ' velocita' di comunicazione RS 232

$romstart = &H2050 ' indirizzo di partenza del codice macchina
$ramstart = &H6000 ' indirizzo di partenza dell'area dati
$ramsize = &H1FFF  ' pongo a 8K l'area dati
$large      ' indirizzamento a 16 bit
***** Inizio programma *****
Waitms 1    ' ritardo per assestamento segnali
.....     ' main del programma
.....
End         ' fine del programma
.....     ' eventualmente procedure del programma

```

L'istruzione di attesa Waitms 1 **è sempre necessaria**. Se non vedete partire il programma appena scaricato controllate la presenza e la posizione di tale istruzione.

GENERAZIONE DEI PROGRAMMI DEFINITIVI

Tutte le considerazioni fatte per la generazione dei programmi di prova valgono anche per la generazione dell'eseguibile definitivo.

Cambia la destinazione, ovvero il programma dovrà essere scaricato e poi scritto nella FLASH **opzionale**, se presente, e quindi cambiano le direttive di compilazioni e la procedura di scaricamento.

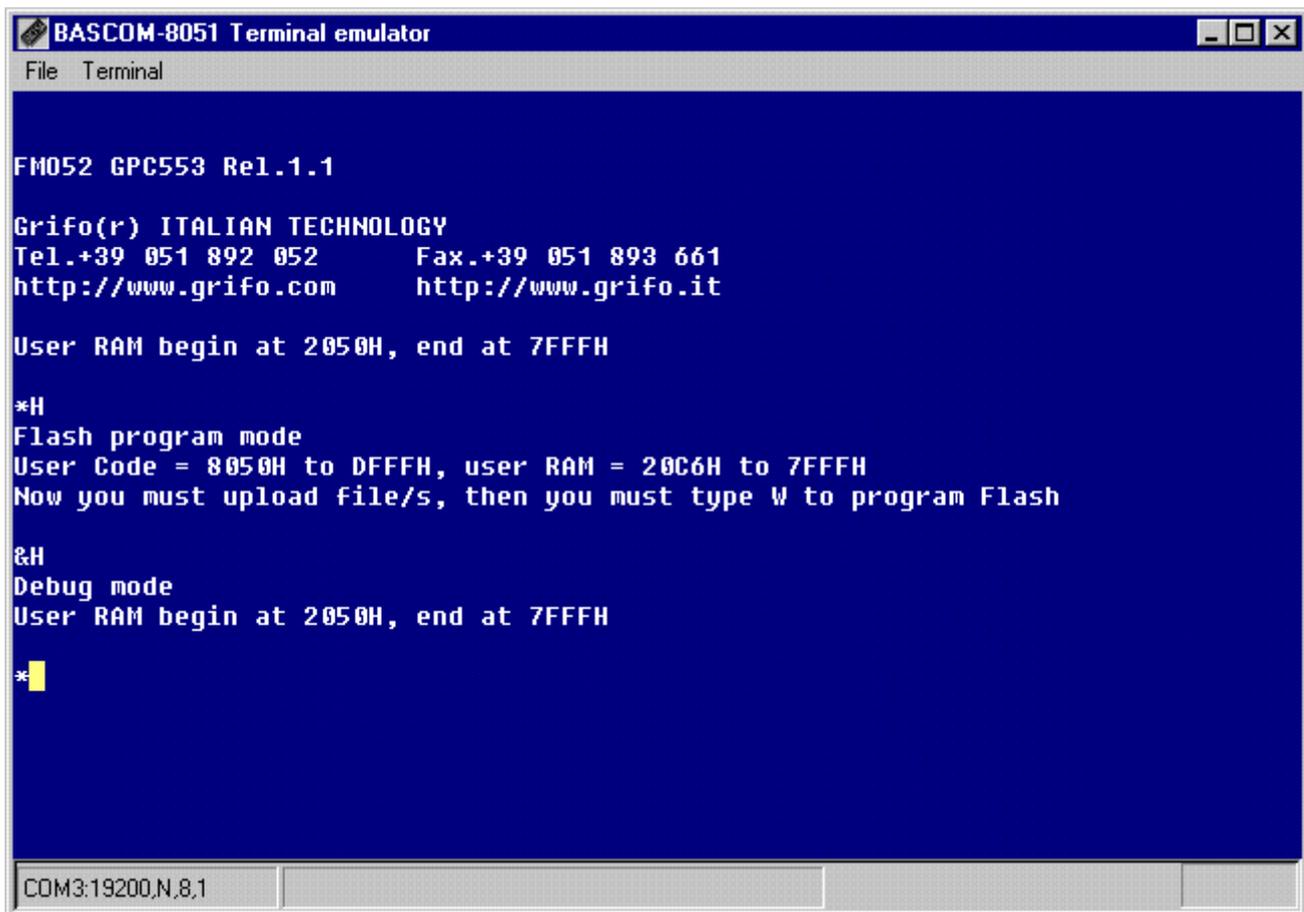
Come si vede dalla mappa della memoria in figura 1, la FLASH si trova mappata dall'indirizzo 8000H fino ad un indirizzo finale che dipende dalla scheda destinazione, per cui per conoscerlo si deve consultare il manuale della GPC® usata.

Nel sorgente **BASCOM 8051** bisogna modificare così le direttive \$romstart, \$ramstart e \$ramsize:

```
$romstart = &H8050   ' indirizzo di partenza del codice macchina
$ramstart  = &H20C6   ' indirizzo di partenza dell'area dati
$ramsize   = &H5EFF   ' pongo a 24K l'area dati
```

In questo caso i valori dati a \$ramstart e \$ramsize sono precisi e vanno copiati così come sono.

Fatta questa modifica è possibile ricompilare il programma, anche qui le operazioni da fare sono le stesse indicate nei capitoli di “UTILIZZO DELL'FMO 52” tranne le ultime due:



```

BASCOM-8051 Terminal emulator
File Terminal

FM052 GPC553 Re1.1.1

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

User RAM begin at 2050H, end at 7FFFH

*H
Flash program mode
User Code = 8050H to DFFFH, user RAM = 20C6H to 7FFFH
Now you must upload file/s, then you must type W to program Flash

&H
Debug mode
User RAM begin at 2050H, end at 7FFFH

*
    
```

FIGURA 11: INGRESSO NELLA MODALITÀ DI PROGRAMMAZIONE DELLA FLASH

UTILIZZO DELLA FLASH DA PROGRAMMA UTENTE

L'area di memoria contenuta nella Flash è suddivisa in 3 parti :

da 8000H a 804FH	Riservata
da 8050H a DFFFH	Programma utente
da E000H a FEFFH	Memoria libera, suddivisa in 124 blocchi di 64 bytes

Quest'ultima area può essere gestita dall'utente a blocchi di 64 bytes durante la modalità di AUTO RUN, tramite l'aiuto di una **CALL** che risponde all'indirizzo 1A00H la quale utilizza 12 bytes di stack.

L'indirizzo di fine della memoria varia da scheda a scheda.

Per il corretto uso della procedura prima della sua esecuzione è necessario inizializzare alcune locazioni di memoria così suddivise :

da 2001H a 2042H	64 bytes che verranno memorizzati in Flash
2043H	indica il numero del blocco, da 0 a 7BH.
2044H	Check sum di tutti i dati, che consiste nel sommare tutti i valori delle locazioni da 2001H a 2043H senza riporto, <u>aggiungendo all fine 1.</u>

Dopo l'uso di questa procedura è indispensabile verificare il contenuto della locazione 2043H, dove e' indicato l'esito della programmazione secondo il valore assunto da essa :

Se = AAH	Numero blocco non valido
Se = BBH	Check-sum errato
Se = EEH	Flash malfunction
Se = 80H	Tutto OK, operazione eseguita.

Ad esempio, per scrivere nel blocco 1 della Flash tutte le 64 locazioni con il valore 00H si devono scrivere nei seguenti indirizzi i corrispettivi valori:

Dati:	da 2001H a 2042H	00H 00H
Blocco:	2043H	01H
Check-sum:	2044H	02H

INTERRUPT

FMO 52 ridireziona tutti i vettori d'interrupt al corrispondente indirizzo all'inizio dell'area SRAM a disposizione dell'utente, fatta eccezione per quello del **TIMER1**, che viene usato internamente per l'esecuzione passo passo ed occupa altre 3 locazioni di memoria SRAM, vedi comando 'S'.

Quando si verifica un programma che usa i vettori d'interrupt è sufficiente:

- allocare quest'ultimi nella SRAM utente, se la modalita' AUTO RUN non è selezionata, ovvero aggiungergli il valore di inizio di tale SRAM (2050H)
- allocare quest'ultimi all'inizio dell'area FLASH per programma utente, se la modalita' AUTO RUN è selezionata, ovvero aggiungergli il valore di inizio di tale FLASH (8050H)

Le procedure di risposta all'interrupt invece non subiscono alcuna variazione.

Utilizzando **BASCOM 8051** ovviamente, l'allocazione dei vettori viene fatta automaticamente dal compilatore e il programmatore non deve preoccuparsene.

Quando viene eseguito il programma utente in modalita' di AUTO RUN, l'**FMO52** prima di eseguire il codice copia i vettori di interrupt dalla Flash alla SRAM, e precisamente da 8050H a 80C5H in 2050H fino 20C5H, in questo modo si avrà lo stesso funzionamento che si aveva nella modalità di debugging.

COMANDI DI FMO 52

RAPPRESENTAZIONE

Al fine di avere la massima quantità di informazioni su una videata del P.C. di sviluppo, durante la fase di debug, l'**FMO 52** è stato progettato per ridurre al minimo le linee di rappresentazione per ogni comando. Tutti i comandi che rappresentano solo una riga di dati (come 'B' e 'S') lo fanno sulla stessa riga su cui il comando è stato fornito (alla sua destra).

Per i comandi che visualizzano grosse quantità di dati (come 'D', 'U', ecc.) premendo lo 'SPAZIO' durante la rappresentazione, questa si fermerà sulla prossima riga. Ogni successiva pressione dello 'SPAZIO' rappresenterà una ulteriore riga mentre la pressione di 'ENTER' riprenderà la rappresentazione a piena velocità delle righe rimanenti.

Il tasto 'ESC' può essere utilizzato per interrompere definitivamente la rappresentazione di un comando, tornando quindi nella modalità di attesa comandi, segnalata dalla presenza di un '*' o '&' a capo riga.

ELENCO

A <indirizzo>

Altera il contenuto della SRAM interna del microcontrollore. **FMO 52** presenta l'indirizzo specificato ed il suo attuale contenuto. Si possono digitare due cifre HEX per cambiare il valore, lo 'SPAZIO' per avanzare alla prossima locazione, il 'BACKSPACE' per ritornare alla locazione precedente o 'RETURN' per terminare il comando di alterazione.

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

Setta un punto d'arresto all'indirizzo specificato. Se l'indirizzo specificato è 0000 il punto d'arresto è rimosso. Ci possono essere fino a quattro punti d'arresto, numerati da 0 a 3.

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

NOTA: La famiglia 8051 di microprocessori non ha una istruzione di salto che occupi un singolo byte e quindi la gestione dei punti d'interruzione non può essere gestita con le modalità classiche. I punti d'arresto sono quindi gestiti inserendo nel codice un'istruzione "LCALL" a seguito dell'esecuzione con il comando "G" e ripristinando il codice originario quando il punto d'arresto viene raggiunto. Ogni '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 e' 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' 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 rimuovere il punto d'arresto raggiunto oppure usare il comando 'S' per avanzare il puntatore programma (PC) fino a sorpassare le locazioni occupate dal punto d'arresto.

C <nome registro> <valore>

Cambia i valori dei registri interni dell'8051.

Il nome registro coincide con una singola lettera ed il valore puo' 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.

D <indirizzo inizio>,<indirizzo fine>

Visualizza il contenuto della memoria dati esterna compresa tra gli indirizzi specificati, in formato Intel Hex ed ASCII. Se come indirizzo di fine e' premuto lo "SPAZIO" allora si assume un indirizzo di fine = FFFF.

E <indirizzo>

Modifica il contenuto dell'area dati esterna del microcontrollore.

FMO 52 presenta l'indirizzo specificato ed il suo attuale contenuto. 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 'RETURN' per terminare il comando di modifica.

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

Riempie la memoria dati esterna dall'indirizzo di inizio fino all'indirizzo di fine con il valore HEX specificato.

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).

H

Abilita l'utilizzo dei comandi dedicati all'uso della Flash, questa modalita' viene indicata dal prompt, se viene rappresentato '*' e' in uso la modalita' normale di monitor, mentre se viene rappresentato '&' sono abilitati i comandi per la Flash (questa modalita' modifica il comando 'L') :

W Programmazione della Flash, copiando il contenuto della SRAM da 2050H fino 7FFFH in Flash da 8050H fino DFFFH.

X Avviene la cancellazione totale del contenuto della Flash.

I <indirizzo inizio>,<indirizzo fine>

Mostra il contenuto della memoria interna compresa tra i due indirizzi specificati, in formato HEX.

L

Attiva la ricezione di un file in formato Intel Hex o S MOTOROLA dal P.C. di sviluppo. Il comando termina in corrispondenza della ricezione delle sequenze ':00' o 'S9' che normalmente coincidono con l'ultimo record dei file.

Se involontariamente viene fornito questo comando, l'utente puo' ritornare al modo comandi digitando direttamente una delle due sequenze descritte.

Se il prompt in uso e' '*', il codice scaricato serve per la modalita' debugger, mentre se il prompt e' '&', il codice scaricato servira' per la programmazione della Flash, infatti durante lo scaricamento il file subira' una riallocazione di 6000H bytes in meno (offset che separa la SRAM dalla Flash), questo permettera' la programmazione della Flash.

All'accensione avvera' l'esecuzione del programma utente direttamente dalla Flash se era attivata la modalita' di **AUTO RUN**.

R

Mostra i valori attuali dei registri dell'8051 (A, B, DPTR, SP, PC, PSW e R0-R7).

S

Esegue una singola istruzione salvata nell'indirizzo attualmente salvato nel puntatore programma (PC). Il disassemblato dell'istruzione eseguita è rappresentato sul P.C. di sviluppo.

L'uso di questo comando comporta la perdita dell'interrupt del TIMER1 overflow, in quanto viene ridirezionato nelle locazioni di memoria RAM: 206BH, 206CH e 206DH.

N.B.: 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 in cui è stato lanciato il comando.

T

Avviene un test completo della RAM a partire dall'indirizzo 2050H fino a 7FFFH. Questo comando altera completamente il contenuto della SRAM, si consiglia, dopo l'uso di questo comando formattare la SRAM con il valore 00H tramite il comando F. (Es: F 2050 7FFF 00)

U <indirizzo fine>,<indirizzo fine>

Disassembla la memoria programma compresa tra i due indirizzi specificati.

Se lo 'SPAZIO' è premuto come indirizzo di fine, si assume il valore FFFF.

Il comando rappresenta l'indirizzo, i byte di codice, il mnemonico e gli operandi di ogni istruzione.

?

Mostre un breve sommario dei comandi disponibili.



APPENDICE A: INDICE ANALITICO

SIMBOLI

\$RAMSIZE 14
\$RAMSTART 14
\$ROMSTART 14
.FMO52 2
.FMO52.32KF 2
/PSEN 2
/RD 2

A

ADDS-VIEWPOINT 10
AT29C256 2
AUTO RUN 4, 17

B

BASCOM 8051 6, 7, 14, 19
BAUD RATE 6, 7, 10
BIT 10

C

COMANDI 20
? 23
A 20
B 20
C 21
D 21
E 21
F 22
G 22
H 22
I 22
L 22
R 23
RAPPRESENTAZIONE 20
S 23
T 23
U 23
COMPILAZIONE 14

D

DATI 6, 7
DEBUG 4

E

EPROM 4

F

FREQUENZA 14

G

GET 51 10

GPC® 5, 15

grifo® 2

I

INTEL HEX 8, 22

INTERRUPT 19

INTRODUZIONE 1

J

JUMPERS 4

M

MAPPAGGIO 4

MCS BASIC 10

MEMORIA 14

P

PARITÀ 6, 7, 10

S

S MOTOROLA 22

SCHEMA 4

SELEZIONE 4

SRAM 4, 15, 20

STOP 6, 7, 10

T

TERMINALE 6

W

WAITMS 15