

## FBasic VI.0

di Giacomo Filippini - Maresca, Pistoia

Il Basic residente nel C-64 è ormai da molto tempo diventato obsoleto nei confronti di quello disponibile nella maggior parte dei computer concorrenti. Fortunatamente esiste la possibilità di migliorarlo attraverso l'utilizzo di vari tool, che gli aggiungono nuove istruzioni finalizzate agli scopi più diversi.

FBasic fa parte della famiglia di utility di espansione Basic: con le sue 45 istruzioni offre grandi vantaggi nella gestione di grafici in alta risoluzione, caratteri programmabili, sprite e disk drive. In particolare mette a disposizione 9 comandi per l'alta risoluzione, 14 per lo schermo e i caratteri programmabili, 6 per la manipolazione degli sprite, 7 per il disk drive, 7 per l'ampliamento delle funzioni Basic ed infine 2 comandi per il corretto funzionamento di FBASIC stesso.

Il programma è totalmente scritto in linguaggio macchina, è allocato a partire dall'indirizzo 49152 (C000) e si estende per 4Kb arrivando all'indirizzo 53247 (CFFF). Dopo il caricamento con LOAD "FBASIC", 8,1 si avvia con un semplice RUN.

La RAM Basic rimane completamente libera, comunque non è consigliabile utilizzare locazioni in pagina zero pena la sovrascrittura dei dati.

L'unica zona non utilizzata da FBASIC è il buffer di cassetta ossia qui 191 Byte a partire dall'indirizzo 828 (033C).

Appena la routine d'entrata viene chiamata, i puntatori in pagina zero vengono alterati in modo da indirizzarli a routine predisposte per la gestione del FBASIC.

In particolare sono deviate le routine di codifica delle parole chiave del BASIC, stampa delle parole chiave gestione dell'emissione di una nuova linea BASIC e la routine di valutazione dei codici BASIC.

La manipolazione di queste routine permette la gestione dei nuovi comandi; infatti al momento dell'immissione di una nuova linea Basic se vi sono presenti comandi di FBASIC, questi verranno memorizzati utilizzando un codice come avviene per il Basic normale del 64.

Questo ci permette di consumare molta memoria in meno.

La gestione dei tasti funzione avviene mediante l'IRQ, per questo anche questa routine viene deviata; anche la routine di NMI ha una sua corrispondente in FBASIC in quanto serve per controllare la pressione di RUN STOP/RESTORE: se quest'ultima condizione si verifica, FBASIC si porta nella condizione iniziale, un po' come succede quando il C-64 si inzializza.

Ci sarebbero molte altre nozioni tecniche, ma non credo sia necessario descriverle in quanto sono implicite nella programmazione in linguaggio macchina.

### Comandi per la bassa risoluzione grafica (modo testo)

CLS

Vuota il video e setta il colore dei caratteri leggendo la locazione 646.

PAPER c (,c1)

Seleziona il colore dello sfondo «c» e se specificato quello del bordo «c1». Tutte le variabili che esprimono un colore devono essere comprese fra il valore zero ed il valore quindici.

INK

Seleziona il colore dei caratteri alterando la locazione 646.

LOCATE x,y, (stringa o espressione)

Posiziona il cursore per la stampa alle coordinate in bassa risoluzione «x» ed «y» che identificano rispettivamente le colonne e le righe.

La coordinata «y» deve essere compresa fra zero e ventiquattro.

La coordinata «x» deve essere compresa fra zero e trentanove.

CENTER stringa

Centra una stringa di caratteri sulla linea di testo in cui si trova il cursore e stampa un RETURN. La stringa può essere lunga da uno a trentotto caratteri.

Se nella stringa sono presenti codici video usati ad esempio per il colore, faranno dal punto di vista pratico la loro funzione, ma da CENTER saranno interpretati come caratteri e per questo, dato che non sono effettivamente visualizzati, sposteranno la scritta che non risulterà più nel centro.

EXTEND zero/EXTEND uno,c1,c2,c3

Rispettivamente disabilita il modo a colore di fondo esteso e lo abilita utilizzando i colori «c1» «c2» «c3».

I caratteri il cui codice di schermo è compreso fra sessantaquattro e centoventisette utilizzano «c1» come colore di fondo esteso, quelli con codice fra centoventotto e centonovantuno utilizzano «c2», mentre quelli con codice fra centonovantadue e duecentocinquanta-cinque utilizzano «c3». Il modo a colore di fondo esteso può essere utilizzato solo in modo testo.

Se «c1» o gli altri colori sono omessi, verranno utilizzati quelli selezionati in precedenza o quelli di DEFAULT che il computer trova nelle locazioni corrispondenti.

I prossimi comandi sono seguiti dal termine (FINESTRA) con cui si intende x,y,x1,y1. Questo perché funzionano in una finestra video determinata appunto nel modo x,y,x1,y1 dove x,y sono le coordinate del carattere in alto a sinistra e x1,y1 quelle del carattere in basso a destra.

Le variabili possono assumere solo determinati valori (come LOCATE).

FILL (FINESTRA), stringa o codifica, c0

Riempie la finestra con il carattere specificato da una stringa o da un numero che identifica il suo codice di schermo colorandolo con il colore «c0» se è specificato; altrimenti usa il colore dei caratteri correnti.

Se il carattere è specificato da una stringa nulla, non viene stampato nulla,

È disponibile, presso la redazione, il disco con i programmi pubblicati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 279

ed è come se l'istruzione non esistesse; attenzione dunque.

LBOX (FINESTRA), carattere, c0

In questo caso, l'istruzione agisce sui bordi interni della finestra disegnando quindi un quadrato o un rettangolo con il carattere espresso del colore «c0». (Per le caratteristiche dei parametri far riferimento al comando FILL sopra descritto).

INVERSE (FINESTRA), i, carattere

Cerca il carattere specificato all'interno della figura e, quando lo trova, lo mette in modo inverse se la variabile «i» è uno e viceversa se la variabile «i» è zero.

Se la stringa è nulla la finestra viene completamente cambiata a seconda del valore della variabile «i» come abbiamo sopra descritto; se il carattere invece è descritto da un codice zero ciò non avviene.

CHANGE (FINESTRA), carattere 1, carattere 2

Questo comando ha molte funzioni a seconda di come si esprimono le variabili «carattere». Se sono tutte e due delle stringhe, viene scambiato nella finestra il carattere uno con il carattere due; se invece il secondo è un numero minore di sedici, viene interpretato come colore e per questo nella finestra sarà cambiato il colore del carattere specificato se esso è presente.

Se sia il primo valore che il secondo sono dei numeri minori di sedici, vengono interpretati come colori e per questo se all'interno della finestra c'è un carattere di colore uno, cambierà in colore due.

Nel caso che siano due numeri, sia CAR.1 che CAR.2, entrambi maggiori di sedici, verranno interpretati come caratteri e avverrà ciò che si era verificato nel primo caso con le due stringhe.

Data la complessità del comando, inserisco alcuni esempi.

**CHANGE** 10,10,20,20, «A», «B»

In questo caso tutti i caratteri «A» presenti nella finestra verranno cambiati in «B»

**CHANGE** 10,10,20,20, «A», 66

Il carattere «A» verrà modificato nell'equivalente carattere di codice scher-

mo 66. In questo caso siamo di fronte ad un numero, che però è maggiore di 16 e per questo è considerato un carattere

**CHANGE** 10,10,20,20, «A», 2

Il carattere «A» assumerà (sempre all'interno della finestra) il colore Rosso (2).

**CHANGE** 10,10,20,20,2,1

Il carattere il quale colore è Rosso (2) diventerà di colore Bianco (1).

**CHANGE** 10,10,20,20,160,42

Il carattere il quale codice è 160 diventerà un asterisco (codice 42) insomma se si inserisce un numero, se questo è maggiore o uguale a sedici sarà considerato un carattere, se non sarà considerato un colore. C'è una condizione per la quale si ottiene un errore

**CHANGE** (finestra), 2, stringa o numero maggiore uguale a sedici. Infatti non è prevista alcuna funzione se si specifica prima il colore e poi il carattere che dovrebbe essere sostituito.

Se le stringhe sono nulle l'istruzione non produce alcun effetto.

SCROLL (FINESTRA), m, carattere, c0

Se «m» è uno si ottiene uno scroll a sinistra della finestra, ovviamente; se invece «m» è zero si ottiene uno scroll a destra.

Il carattere può essere definito da una stringa o da un numero come gli altri comandi di questo tipo; se è nulla, viene scrollata la finestra riportando sia i colori che i caratteri che sono usciti; se invece il carattere è specificato, al posto dei caratteri usciti entrerà quest'ultimo, colorato con «c0» se è specificato altrimenti del famoso colore di DEFAULT come abbiamo già visto in precedenza.

### Comandi per l'alta risoluzione grafica

HGR c0, c1

Attiva la pagina grafica in modo normale settando i colori «c0» di fondo e «c1» di plottaggio. La pagina grafica viene posta

«sotto» la ROM del Kernal a partire dall'indirizzo 57334 (E000)

CLEAR

Cancela la pagina grafica

TEXT

Attiva il modo testo; e si usa per passare dalla visualizzazione in alta risoluzione a quella in modo testo

COLOR n

Seleziona il colore di plottaggio; «n» può assumere quattro valori che indicano il colore scelto nel modo:

0 - Colore di sfondo usato per cancellare determinati punti.

1 - Colore di plottaggio rispetto alla coppia di bit (11).

2 - Colore di plottaggio rispetto alla coppia di bit (10).

3 - Colore di plottaggio rispetto alla coppia di bit (01).

Quando la pagina grafica è in modo normale consigliamo di utilizzare i colori 0 e 1; infatti gli altri due sono definiti per il modo multicolore e vengono perciò settati dall'istruzione MULTI.

MULTI n ((c0),c1)

Quando la variabile «n» assume il valore uno, viene settato il modo Multicolore e vengono definiti i colori di estensione.

Se non sono specificati i colori, verranno utilizzati quelli segnalati dalle locazioni 53282 e 53283 oppure se erano già stati definiti verranno utilizzati quest'ultimi contenuti comunque nelle stesse locazioni.

Quando «n» invece è uguale a zero il modo Multicolore viene disabilitato.

PLOT x,y

Disegna un punto definito dalle coordinate x ed y.

«x» può assumere valori da zero a centocinquantanove ed «y» da zero a centonovantanove; sia che si lavori in modo normale o multicolore.

LINE x,y TO x1,y1

Disegna una linea dal punto di coordinate x,y al punto di coordinate x1, y1 nel corrente colore di plottaggio.

BOX x,y,x1,y1,

Disegna un rettangolo del quale il punto in alto a sinistra a coordinate x,y e quello in basso a destra a coordinate x1, y1.

CHAR x,y, «testo», INVERSE, SET

*Si prega il Signor Luciano Rosa di mettersi in contatto con la redazione.*

Permette di scrivere dei testi in pagina grafica, i codici che normalmente si inseriscono nell'istruzione PRINT non vengono considerati.

Le variabili x ed y identificano la posizione di stampa come l'istruzione LOCATE, infatti x deve essere compreso fra zero e trentanove ed y invece tra zero e ventiquattro.

La variabile inverse, può assumere il valore zero se il testo deve essere stampato in modo normale, ed invece uno se il testo lo si richiede in modo inverse.

La variabile set, identifica il set di caratteri che si vuole utilizzare in stampa, può assumere tre valori: zero per il set maiuscolo, uno per il set minuscolo, due per il set ridefinito allocato all'indirizzo 14336 come si ottiene con l'istruzione CSET.

In pagina grafica possono comparire contemporaneamente caratteri riferiti ai diversi set. Anche se non sono attivati, perché i dati relativi alle matrici dei caratteri sono letti direttamente dalla memoria. Se si pretende di usare il set ridefinito senza averlo prima ricopiato con l'istruzione CSET verranno visualizzati caratteri privi di significato.

### Comandi per la gestione degli SPRITE

SPRITE n,b1,c0

Definisce uno sprite di numero «n» (0-7); i dati che ne definiscono la forma sono contenuti nel blocco «b1» (0-255) ed il suo colore sarà «c0» (0-15).

Se si vuole definire più di uno sprite basta inserire dopo i tre dati una virgola e dopo eseguire tutto come un'istruzione normale; esempio:

SPRITE 1,14,2, 2,24,3

Questa regola vale per tutti i comandi utilizzati per gli sprite eccetto SMULTI che non ne ha bisogno.

SMULTI c0, c1

Definisce i colori per il multicolore degli sprite.

ASPRITE n

Attiva lo sprite «n». Possono essere inseriti altri numeri di sprite e verranno anch'essi attivati (vedi SPRITE).

DSPRITE n

Disattiva lo sprite «n»; anche per questo comando vale la legge descritta per il comando (SPRITE).

SATTR n,xesp, yesp, prior, multi

Definisce gli attributi delle sprite «n».

Per attivare un parametro si inserisca un uno, per disattivarlo uno zero. XESP definisce l'espansione orizzontale; YESP l'espansione verticale; PRIOR definisce la priorità sui caratteri dello schermo e MULTI definisce se lo sprite è multicolore oppure no.

MOVE n,x,y

Posiziona lo sprite «n» alle coordinate

x,y dove x può assumere i valori da zero a cinquecentoundici; e y da zero a duecentocinquantacinque. Comunque quando questi superano certi valori lo sprite non viene visualizzato perché fuori dallo schermo.

### Comandi per la gestione dei caratteri programmabili

CSET n

Copia il set descritto da «n» a partire dall'indirizzo 14336. Quando «n» è zero vengono copiati i caratteri maiuscoli, invece quando «n» è uno vengono copiati i caratteri minuscoli.

Ho scelto di limitare un po' questa funzione invece che le altre in quanto variando i banchi di memoria per allocare diversamente i caratteri si varierebbe anche la posizione del buffer video, e per questo le istruzioni più complicate non funzionerebbero.

Per questo ho scelto di rilocare solo un set ed arbitrariamente all'indirizzo 14336.

Ci sarebbe stata un'altra soluzione; ossia l'indirizzo 12288, ma poi il Basic sarebbe penalizzato troppo dal punto di vista di memoria libera, per questo infine ho scelto il sistema sopra descritto in modo da lasciare più spazio libero ed avere però un set di caratteri utilizzabili. Consigliamo quando si usa CSET di limitare la memoria Basic agendo sulle locazioni utili a questo scopo.

ASET

Questa istruzione attiva il set.

Non va usata quando la pagina in alta risoluzione è attivata.

DSET

Disabilita il set di caratteri programmabili.

Non va usata quando la pagina in alta risoluzione è attivata.

### Comandi per migliorare funzioni Basic

DOKE IN,VAL

Esegue una doppia POKE all'indirizzo «in».

In parole povere è la famosa POKE a sedici bit, infatti «val» può essere un numero maggiore di 255.

DEEK IN, variabile che acquisisce il risultato.

Esegue una doppia PEEK all'indirizzo «in» ossia legge il valore contenuto in «in» e lo considera BASSO (LOW), poi legge il valore contenuto in «in+1» e lo considera ALTO (HIGH) ottenuti questi valori restituisce il numero a sedici bit corrispondente.

È molto differente il funzionamento di questa istruzione da quella normale in quanto la variabile che prende il risultato è segnalata dopo l'indirizzo di lettura;

perciò fare molta attenzione. Se la variabile non deve essere specificata, il risultato verrà stampato sul video.

JOY n,v0, v1

La variabile «n» può assumere due valori rispettivi alla porta joystick che si vuole leggere.

Le variabili che seguono «n», sono le variabili nelle quali il computer memorizza rispettivamente il risultato sullo spostamento della barra; e il test sul pulsante di FUOCO.

«v0» assumerà i seguenti valori a seconda della posizione della barra.

```
      1
     5 7
    3 0 4
     6 8
      2
```

Invece «v1» assumerà zero se il pulsante non è premuto ed uno nel caso contrario.

LRES n

Esegue un RESTORE alla linea specificata «n».

Se la linea non esiste il computer segnalerà l'errore.

NUMBER (STRINGA),VAR.

Converte il numero esadecimale contenuto nella stringa ed il risultato lo immagazzina nella variabile «var».

Il numero esadecimale può essere del tipo (\$ nn oppure \$ nnnn) dove «nn» sono le cifre esadecimali. Il dollaro non deve essere omissso.

NUMBER (VAR), STRINGA

Questa è la seconda funzione di NUMBER, infatti converte in ESA.

Il numero contenuto nella variabile «var» e assegna il risultato alla stringa specificata. Inserisce prima del numero il famoso dollaro.

KEY n, «caratteri»

Definisce il tasto funzione «n» assegnandogli la stringa di caratteri specificata.

I caratteri non devono essere maggiori di 14.

Se si volessero inserire degli apici al loro posto per ovviare problemi, basta inserire l'accento (') e se si vuol inserire anche un codice RETURN basta digitare la freccia a SINISTRA.

Esempio:

LOAD'ABC', 8 ←

Alla pressione del tasto al quale è associato questo gruppo di caratteri otteniamo il caricamento del programma ABC.

Il comando KEY come qui specificato funziona in modo diretto. All'interno di un programma ha la seguente funzione molto diversa da quella sopra descritta.

Se si inserisce un comando KEY senza parametri, appena l'interprete lo legge aspetta la pressione di un qualsiasi tasto.

Se invece è della forma: KEY, «a»; si ottiene una pausa finché il tasto «a» non viene premuto. Se la stringa ha più di un carattere, il computer li attenderà uno per uno; attenzione che nella stringa non compaiono particolari codici come quelli del colore, perché in questo caso il computer aspetterà la pressione dei due tasti che identificano il codice; ad esempio CTRL-2.

Vari tipi di sintassi /KEY/KEY, «a»/KEY, «abc»/KEY, «A», «B», «C».

La terza e la quarta svolgono la stessa funzione. Se si entra nel programma mediante una GOTO, ci saranno problemi con questa istruzione dato che per il computer siamo ancora in modo diretto (loc. 157).

GCAL (ESPRESIONE NUMERICA)

Esegue una GOTO all'indirizzo segnalato dal risultato dell'espressione. Far attenzione al caso che si usi una routine di rinumerazione delle linee; perché sicuramente questa istruzione non verrà cambiata; agite quindi di conseguenza.

### Comandi per la gestione del drive

DIR

Visualizza la directory del dischetto presente nel drive senza alterare il programma Basic residente in memoria.

ERR

Visualizza lo stato del drive, questa istruzione va usata quando la spia del drive lampeggia perché si è verificato un errore.

DISK «comando:nome»

Invia attraverso il canale 15 delle istruzioni al drive; è l'equivalente della forma OPEN 15,8,15, «comando:nome»:close 15.

DEL «nome»

Cancella dal disco il programma «nome».

RESV «nome»

Questo comando, cancella prima di salvare il programma «nome», con questo, si scavalca il problema del «save con replace» ottenuto con l'istruzione SAVE «C:nome» in quanto quest'ultima porta dei problemi quando lo spazio sul disco è poco, perché prima salva e poi cancella; invece come abbiamo detto RESV prima cancella e poi salva così da non cadere in certi errori che potrebbe dare il vecchio sistema, comunque funzionano tutti e due i modi.

BSV «nome», INDIRIZZO D'INIZIO, INDIRIZZO DI FINE.

Questo comando permette di salvare su disco determinate zone della memoria identificate dai parametri da inserire dopo il nome, è consigliabile salvare un BYTE in più del dovuto così da esser certi che la registrazione è corretta.

BLD «nome», indirizzo d'inizio

Va utilizzato quando si vuol caricare all'indirizzo specificato un file di dati anche se quest'ultimi sono stati salvati ad un indirizzo diverso da quello segnalato.

Può esser vantaggioso quando si vuol caricare un set di caratteri da utilizzare con FBasic, infatti basta inserire come indirizzo d'inizio il valore 14336.

Questa istruzione non dà problemi di errori di OUT OF MEMORY che si riscontrano invece nel caricamento di dati da programma nel modo normale.

### Comandi per la gestione di FBasic

IRQ

Se si lavora con qualche routine in linguaggio macchina mentre FBasic è in memoria e sta lavorando, può succedere che il microprocessore incontri una istruzione BRK, se questo accadesse il computer verrebbe inizializzato, e i puntatori degli interrupt riportati al loro originario

valore. Questo fatto vi priverebbe dei tasti funzione, ed è per questo che è stato necessario inserire un comando per la riabilitazione degli interrupt di FBasic, e non si poteva chiamare altro che IRQ. QUIT

Disabilita Fbasic riportando il computer al modo normale ossia con il suo Basic senza espansione.

Se si volesse rientrare in FBasic, occorrerebbe digitare: SYS 49152 (RETURN).

FBasic non è molto compatibile con l'istruzione THEN.

Una sintassi del tipo:

IF A=B THEN DIR

genererebbe un errore di sintassi.

Per ovviare a questo inconveniente occorre inserire fra la THEN e il comando successivo, se questo è di FBasic, i due punti così da ottenere la sintassi:

IF A=B THEN: DIR

Questo accorgimento permette di scavalcare l'errore di sintassi.

## TOD-Basic

di Gaetano Minardi - Niscemi (CL)

Tutti sanno che il C-64 possiede un orologio quasi perfetto al quale è possibile accedere tramite le variabili TIS e TI.

Ogni qual volta si accende il computer l'orologio viene attivato per cui richiamando le variabili TI e TIS è possibile conoscere con buona precisione da quanto tempo il nostro C-64 è acceso. Ma le suddette variabili possono pure essere controllate in modo da fare segnare all'orologio l'ora desiderata. Con le procedure usuali non è possibile avere un orologio sempre presente sullo schermo che funzioni in modo continuo mentre si sta programmando o mentre è in esecuzione un programma Basic qualsiasi: le variabili vengono considerate infatti come parte integrante di un programma e perciò eseguite singolarmente. Così nel momento in cui queste variabili sono eseguite non è possibile sfruttare il Commodore 64 per altri scopi. Da quanto detto sembrerebbe che realizzare un programma che visualizzi un orologio sullo schermo sia un sogno irrealizzabile. Per ovviare a questo inconveniente ed avere sempre presente sullo schermo un orologio, si può però fare ricorso all'interrupt e quindi al linguaggio macchina. Vediamo, prima di ogni cosa, come funziona l'orologio accessibile tramite le variabili sopra citate. Ogni sessantesimo di secondo si verificano delle interruzioni (ogni cinquantesimo di secondo nelle versioni europee

del C-64). Durante queste interruzioni comandate dal CIA (Complex Interface Adapter) viene chiesto al 6510 di eseguire un lavoro indispensabile al funzionamento del sistema, quale il lampeggio del cursore, la scansione di tastiera e molte altre operazioni tra cui quella di aggiornare l'orologio interno. A volte può capitare di disattivare i tasti RUN-STOP/RESTORE mediante le interruzioni interne, oppure di usare l'interrupt per vari motivi, rischiando così di compromettere il cronometrando delle variabili dell'orologio. Inoltre c'è da dire che durante le operazioni di salvataggio e di caricamento l'aggiornamento dell'orologio viene temporaneamente bloccato. A questi inconvenienti si può ovviare usando l'orologio TOD (Time Of Day) del CIA 6526. Questo orologio non dipende dagli interrupt e può permettersi di funzionare liberamente all'interno del chip 6526. TOD-Basic sfrutta proprio le potenzialità di questo chip. Prima di spiegare il funzionamento del TOD-Basic è meglio dare un'occhiata al CIA. All'interno del C-64 ci sono due CIA: essi, oltre alle funzioni sopra citate, controllano le periferiche collegate alla macchina facendo da mediatori tra il processore e il mondo esterno. La loro funzione è: assai simile, ma mentre il primo controlla precise funzioni della memoria e il bus seriale (dove si allacciano le periferiche come la stampante, il drive, eccetera), l'altra interfaccia controlla la tastiera e ogni cosa collegata alle «control port» 1 e 2. Indubbiamente per poter gestire tutti questi elementi, ogni computer necessita di un

orologio che sincronizzi questi scambi d'informazione.

Per questo motivo in ogni CIA ci sono ben tre orologi, di cui due a sedici bit e uno a otto bit, denominato TOD, completamente identico ai comuni orologi. Mentre gli orologi a sedici bit vengono usati con molta frequenza dal sistema operativo, il TOD sembra completamente trascurato. Il CIA 1 si trova a partire da \$DC00 (56320 decimale), mentre il CIA 2 parte da \$DC00 (56576 decimale). Ogni CIA contiene 16 registri che si ripetono ad incrementi regolari di 16. Così, per esempio, il valore contenuto in \$DC00 comparirà pure in \$DC10, \$DC20, ... eccetera. Il mio programma sfrutta in particolare soltanto quattro di questi registri, e più precisamente: \$08 (8 decimale) - decimi di secondo; \$09 (9) - secondi; \$0A (10) - minuti; \$0B (11) - ore. Quest'ultimo registro è usato per vedere se ci si trova in presenza di un'ora antimeridiana o pomeridiana. Nel registro \$0B è depositato

il flag di AM/PM e se il settimo bit di questo registro è posto a 0, ci si trova in presenza di un'ora antimeridiana (AM) mentre, se è posto a 1, l'orologio sta indicando un'ora pomeridiana (PM). L'ora visualizzata dal programma TOD-Basic sarà accompagnata, per una buona consultazione, dalle sigle AM e PM, a seconda dell'ora introdotta in partenza. C'è da notare che quando l'ora segnata è 11:59:59AM, scattando le ore 12:00:00, la sigla AM sarà sostituita automaticamente da PM.

TOD-Basic si attiva con SYS 49152. Per poter gestire gli orologi, dovranno essere seguite le istruzioni elencate più avanti.

### Descrizione dei comandi

Veniamo ora alla descrizione dei tre comandi che permettono di gestire gli orologi. Sì, parliamo di orologi (quindi al plurale) in quanto è possibile visualizzar-

ne due contemporaneamente (uno nell'angolo superiore sinistro e l'altro nell'angolo superiore destro) e farli lavorare separatamente con valori diversi. Questo è possibile sfruttando contemporaneamente i due CIA 6526.

Ecco i tre comandi che rendono estremamente semplice l'uso di TOD-Basic.

**CLOCK:** permette di attivare gli orologi e visualizzarli sullo schermo.

Sintassi: CLOCK [1/2], [ON/OFF]. Il primo parametro consente di visualizzare l'orologio 1 o 2 sullo schermo; il secondo parametro lo attiva se posto in modalità ON, mentre lo disattiva se viene posto in modo OFF.

Esempi: CLOCK 1 visualizza l'orologio 1 sullo schermo e lo attiva.

CLOCK 1, ON ha la stessa funzione del comando precedente. CLOCK 2, OFF disattiva il secondo orologio cancellandolo dallo schermo.

**TIME:** consente di regolare e attivare un orologio.

Sintassi: TIME [0/1], [ora+ AM/PM]. Il primo parametro indica l'orologio che si intende attivare, mentre il secondo parametro consente di regolare l'ora.

L'ora desiderata deve essere scritta nel formato: ORE/MINUTI/SECONDI (se si vorrà scrivere l'una e due minuti esatti si dovrà scrivere 010200) seguito da AM (se ore antimeridiane) o PM (se ore pomeridiane).

Esempi: TIME 0, "105320 AM" attiverà l'orologio 1 facendogli segnare le ore 10, cinquantatré minuti e venti secondi antimeridiani. TIME 1, "043245 PM" attiverà l'orologio numero 2 facendogli segnare le ore quattro, trentadue minuti e quarantacinque secondi pomeridiani.

**TIMES:** funziona quasi come le variabili stringa TI\$ del Basic standard, solo che in questo caso controlla l'orologio TOD del CIA. Può essere manipolata come una normale variabile del Basic. Esempi: PRINT TIMES\$ (0) stampa il valore del primo orologio; AS\$=TIMES\$(1) stampa il valore corrente del secondo orologio.

Nuovi colori di Default. Una cosa importante del TOD-Basic è che si possono cambiare a piacimento i colori di default del bordo dello schermo e del carattere, regolando opportunamente (mediante le POKE per intenderci) le seguenti locazioni: 49871 (colore bordo), 49872 (colore schermo), 49873 (colore carattere). Esempi: POKE 49871,1; POKE 49872,1; POKE 49873,0 cambierà i colori di default del C-64.

Più precisamente il bordo e lo schermo diventeranno di colore bianco, mentre il carattere di colore nero. Questi nuovi colori potranno essere attivati con la pressione simultanea di RUN/STOP+RESTORE.

#### TOD-Basic

```
10 PRINT"(CLR) CARICAMENTO DATI...":X=49152
20 PRINT"(HOME) (DOWN) (DOWN) LOCAZIONE: ";X
30 READA:IFA=-1THEN50
40 POKEX,A:X=X+1:GOTO20
50 SYS 49152
60 END
10000 DATA076,210,194,076,079,067,203,073,077,197,000,032,115,000,201,067,240
10010 DATA010,201,084,240,034,032,121,000,076,231,167,160,000,200,177,122,056
10020 DATA249,002,192,240,247,201,128,208,235,200,032,251,168,032,121,000,032
10030 DATA081,192,076,108,193,160,000,200,177,122,056,249,006,192,240,247,201
10040 DATA128,208,207,200,032,251,168,032,114,193,076,174,167,032,121,000,208
10050 DATA006,169,000,141,255,002,096,032,158,183,224,003,176,248,224,000,240
10060 DATA239,134,002,032,121,000,240,031,032,253,174,032,121,000,201,145,240
10070 DATA021,032,248,168,173,255,002,166,002,224,001,240,003,041,127,044,041
10080 DATA128,141,255,002,096,173,255,002,166,002,224,001,240,003,009,128,044
10090 DATA009,002,141,255,002,076,248,168,162,000,134,168,169,220,133,252,134
10100 DATA002,169,011,056,229,002,133,251,160,000,166,002,177,251,157,167,002
10110 DATA173,011,220,141,171,002,041,127,141,167,002,166,002,189,167,002,041
10120 DATA240,074,074,074,074,024,105,048,164,168,153,172,002,200,189,167,002
10130 DATA041,015,024,105,048,153,172,002,200,169,058,153,172,002,200,132,168
10140 DATA166,002,232,224,004,208,180,096,173,255,002,041,127,240,036,032,161
10150 DATA192,162,000,189,172,002,157,000,004,232,224,008,208,245,232,169,001
10160 DATA172,171,002,016,003,024,105,015,157,000,004,232,169,013,157,000,004
10170 DATA173,255,002,041,128,240,052,238,189,192,169,221,133,252,162,000,134
10180 DATA168,134,002,032,171,192,162,000,189,172,002,157,028,004,232,224,008
10190 DATA208,245,232,169,001,172,171,002,016,003,024,105,015,157,028,004,232
10200 DATA169,013,157,028,004,206,189,192,076,049,234,120,169,246,141,020,003
10210 DATA169,192,141,021,003,088,096,032,095,193,076,174,167,032,158,183,224
10220 DATA002,144,002,162,001,134,187,032,253,174,032,158,173,032,163,182,201
10230 DATA008,240,003,076,072,178,169,220,024,101,187,133,252,169,009,133,251
10240 DATA160,000,177,034,153,060,003,200,192,008,208,246,162,000,160,002,173
10250 DATA066,003,201,065,240,007,173,060,003,009,008,208,003,189,060,003,010
10260 DATA010,010,010,133,167,232,189,060,003,041,015,005,167,133,167,224,002
10270 DATA176,014,165,167,041,127,201,018,144,006,165,167,073,128,133,167,165
10280 DATA167,145,251,232,136,016,211,198,251,160,000,177,251,145,251,076,248
10290 DATA168,169,000,133,013,032,115,000,176,003,076,243,188,032,019,177,176
10300 DATA003,076,154,174,201,084,240,003,076,040,175,160,000,200,177,122,056
10310 DATA249,006,192,240,247,201,128,208,238,200,177,122,201,036,208,231,152
10320 DATA200,032,251,168,032,250,174,032,158,183,224,002,144,003,076,072,178
10330 DATA032,247,174,134,167,162,000,134,168,134,002,169,220,024,101,167,133
10340 DATA252,141,189,192,120,032,171,192,162,000,189,172,002,157,255,000,232
10350 DATA224,008,208,245,169,077,141,008,001,169,000,141,009,001,169,065,141
10360 DATA007,001,160,003,177,251,016,005,169,080,141,007,001,088,169,220,141
10370 DATA189,192,076,111,180,032,021,253,032,095,193,169,139,141,024,003,169
10380 DATA194,141,025,003,096,072,138,072,152,072,169,127,141,013,221,172,013
10390 DATA221,048,028,032,002,253,208,003,108,002,128,032,188,246,032,225,255
10400 DATA208,012,032,122,194,032,163,253,032,185,194,108,002,160,076,114,254
10410 DATA032,024,229,173,207,194,141,032,208,173,208,194,141,033,208,173,209
10420 DATA194,141,134,002,096,000,000,007,032,122,194,169,011,141,008,003,169
10430 DATA192,141,009,003,169,193,141,011,003,169,238,141,010,003,160,194,169
10440 DATA243,032,030,171,076,123,227,013,084,079,068,032,066,065,083,073,067
10450 DATA032,065,084,084,073,086,065,084,079,013,000,000,000,000,000,000,000
10460 DATA -1
```

READY.



## Elenco del software disponibile su cassetta o minifloppy

Per ovviare alle difficoltà incontrate da molti lettori nella digitazione dei listati pubblicati nelle varie rubriche di software sulla rivista, MCmicrocomputer mette a disposizione i programmi più significativi direttamente su supporto magnetico. Riepiloghiamo qui sotto i programmi disponibili per le varie macchine, ricordando che i titoli non sono previsti per computer diversi da quelli indicati. Il numero della rivista su cui viene descritto ciascun programma è riportato nell'apposita colonna; consigliamo gli interessati di procurarsi i relativi numeri arretrati, eventualmente rivolgendosi al nostro Servizio Arretrati utilizzando il tagliando pubblicato in fondo alla rivista.

Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Carlo Perrier 9, 00157 Roma.

Codice	Titolo Programma	MC n.	Prezzo
<b>APPLE II</b>			
DA2/20	Apple Puzzle II	65	15000
DA2/21	Precisione Multipla	68	15000
DA2/22	Sistema 2 + Toto 5.3 IIGS	68	15000
DA2/23	Operazione Apokalypsis	71	30000
DA2/24	Classifiche di Formula 1	72	15000
DA2/25	Programmabile RPN	73	15000
DA2/26	Supercircle + Polygonale	74	15000
DA2/27	Hard Copy OKI 83/A	76	15000
DA2/28	ProDOS Utility	77	15000
DA2/29	Modulo Base	78	15000
DA2/30	List db	79	15000
DA2/31	Bioritmi	80	15000
<b>COMMODORE AMIGA</b>			
DAM/01	F-15	63	15000
DAM/02	Gest. liste programmi	64	15000
DAM/03	Studio di Funzioni	65	15000
DAM/04	Math Pack	68	15000
DAM/05	Redcode & Mars (Core Wars)	68	15000
DAM/06	Life	69	15000
DAM/07	Rubrica Telefonica	70	15000
DAM/08	Piramidi	70	15000
DAM/09	Regolazione dei colori	71	15000
DAM/10	Analitica	71	15000
DAM/11	Grafici	72	15000
DAM/12	Traduttore	73	15000
DAM/13	La Borsa	74	15000
DAM/14	DMA Music Compiler	74	15000
DAM/15	Poker	78	15000
DAM/16	Programmi per il Copper	79	15000
DAM/17	Mandelbrot mania	81	15000
DAM/18	SF-Search File	86	15000
DAM/19	AMIPac	87	15000
<b>MS-DOS</b>			
DMS/01	Plotter + Morse	67	15000
DMS/02	Meloe + Spawn	68	15000
DMS/03	Pretty + Scritte scorrevoli + Compute	69	15000
DMS/04	Emulatore CGA per Hercules	70	15000
DMS/05	Turbo Directory	71	15000
DMS/06	Math Tool S	72	15000
DMS/07	Bioritmi + Routine	72	15000
DMS/08	Salvavideo + Scritte scorrev. + PG151	73	15000
DMS/09	Optimizer + Indenter dBase III	74	15000
DMS/10	Joystick Controller	75	15000
DMS/11	BootSlow + SlowDown + Turbo Utility	76	15000
DMS/12	Redcode & Mars (Core Wars)	76	15000
DMS/13	Gestione Errori Critici Disco + PosCur	77	15000
DMS/14	Finestre & Desk	78	15000
DMS/15	General Manager	78	15000
DMS/16	Tool 05	79	15000
DMS/17	PullDown Menu + Retrace	80	15000
DMS/18	Righe	81	15000
DMS/19	La spada di Krall	82	15000
DMS/20	Regressione	82	15000
DMS/21	Tesseract + Charset Editor	83	15000
DMS/22	Sega-File + Installatore	84	15000
DMS/23	Shape Tool-kit	87	15000
DMS/24	Starfox	88	15000
DMS/25	Inline in Turbo C + Svuota Keyboard Buffer	89	15000
DMS/26	Sistemi parametrici + GIFDIR 1.0	90	15000
DMS/27	QuickMatrix + Stampa Virtuale	91	15000
<b>ATARI ST</b>			
DST/01	Virus Killer	74	15000
DST/02	Mandelbrot + Proiez. Ort. + Bilancio	78	15000
DST/03	Diagrammi di Henon	81	15000
DST/04	Paroliamo	84	15000
DST/05	Enalotto	85	15000
DST/06	Cariche	83	15000
DST/07	Flow-Chart	87	15000
DST/08	Funz Manual	89	15000
DST/09	RedCode	88	15000
<b>COMMODORE 128</b>			
D28/01	MMCalc	53	15000
D28/02	Hardcopy 128	55	15000
D28/03	Sheet II	57	15000
D28/04	Star Quest	58	15000
D28/05	Family Budget	60	15000
D28/06	La Casa Stregata	61	15000
D28/07	Strutture 80/33	63	15000

Codice	Titolo Programma	MC n.	Prezzo
D28/08	Bas 80 V. 2.0a	64	15000
D28/09	Paint 80 1.0	65	15000
D28/10	Bas 80 V. 2.11	66	15000
D28/11	Calendario Perpetuo + Montecarlo	67	15000
D28/12	Disegna Circuiti	68	15000
D28/13	Mark's Data Base	70	15000
D28/14	Label Disk + Disk Editor + Dem DOS	71	15000
D28/15	Pulldown 128HR + Menu + Drawer	72	15000
D28/16	Prospettive	73	15000
D28/17	Char 80 V. 1.0	74	15000
D28/18	Italia 128	75	15000
D28/19	Super Sprite	77	15000
D28/20	Othello	80	15000
D28/21	Expert System Shell 128	81	15000
D28/22	Kit di programmazione S.O.G.A.R. 128	82	15000
D28/23	Caratteri Programmabili	83	15000
D28/24	Brush & Mouse	84	15000
D28/25	Astrolabio	89	15000
D28/26	Sort Vettore Stringa + Input Stringa	90	15000
D28/27	Piano Cartesiano + Calendario	91	15000
<b>COMMODORE 64</b>			
D64/14	Link-64	57	15000
D64/15	New Char 2.2	58	15000
D64/16	Music 64	59	15000
D64/17	TRX-MEM	59	15000
D64/18	WOS + WBasic	60	15000
D64/19	Strange Basic + Dracula	63	15000
D64/20	File Rescue	64	15000
D64/21	La Casa	64	15000
D64/22	Digital Voice	65	15000
D64/23	Vita 3D	65	15000
D64/24	Corso di Linguistica	66	15000
D64/25	Archipius	66	15000
D64/26	Math Pack Plus	66	15000
D64/27	Scroll + Multitask + Classifica	67	15000
D64/28	Calend. Perpetuo + Effetto Telecamera	68	15000
D64/29	Listing Plus + Utility Data	68	15000
D64/31	Trucchi e Routine per programmatori	71	15000
D64/32	Flow-Chart + Flower's Love	73	15000
D64/33	Sprite Editor	76	15000
D64/34	Portfolio 64 + Elim. bordi schermo	77	15000
D64/37	Intonatore	81	15000
D64/38	Gendista 64	82	15000
D64/40	Block Utility	84	15000
D64/41	Sprite 6499	85	15000
D64/42	Moi Moi Labyrinth + Tool 64	86	15000
D64/43	Titolatore 64	88	15000
D64/44	Editor Musicale	89	15000
D64/45	Dump Variabili	90	15000
D64/46	FBasic V1.0 + Tod-Basic	91	15000
<b>MSX</b>			
DMX/01	Toto 13	60	15000
DMX/02	Painter	62	15000
DMX/03	MSX Bank	63	15000
DMX/04	Grafica 3D + Hard Copy	65	15000
DMX/05	Easy Disk	66	15000
DMX/06	Classifiche	67	15000
DMX/07	Magic Paint	67	15000
DMX/08	Autogest	68	15000
DMX/09	Compilatore v. 1.01	69	15000
DMX/10	Diskmap	70	15000
DMX/11	Mini dBase MSX	71	15000
DMX/12	Grafica in Turbo Pascal	72	15000
DMX/13	Math Pack Plus 3.20	73	15000
DMX/14	QB/CAD	75	15000
DMX/15	Simple Desk	76	15000
DMX/16	The MSX2 Super Print	77	15000
DMX/17	Grafica in Turbo Pascal (Graph 1&2)	77	15000
DMX/18	Hard Copy	78	15000
DMX/19	HEXDUMP	79	15000
DMX/20	Utilities in Turbo Pascal	80	15000
DMX/21	dBase MSX Plus	81	15000
DMX/23	PutChar + 4010 S	84	15000
DMX/24	Analysis + Lister Basic	85	15000
DMX/25	Aliment MSX	86	15000
DMX/26	SortDir + Hochtogonus + 3D Mover	87	15000
DMX/27	Tool/Extended Basic + Bindata Bas	88	15000
DMX/28	Chain & Window + Chain Utility	90	15000
DMX/29	Latino Pack	91	15000

Nota: Il supporto è rappresentato da dischi da 5.25" e 3.5" in relazione alle caratteristiche dei disk drive presenti sul computer in oggetto.

# Basta con le Rigenerazioni in AutoCAD

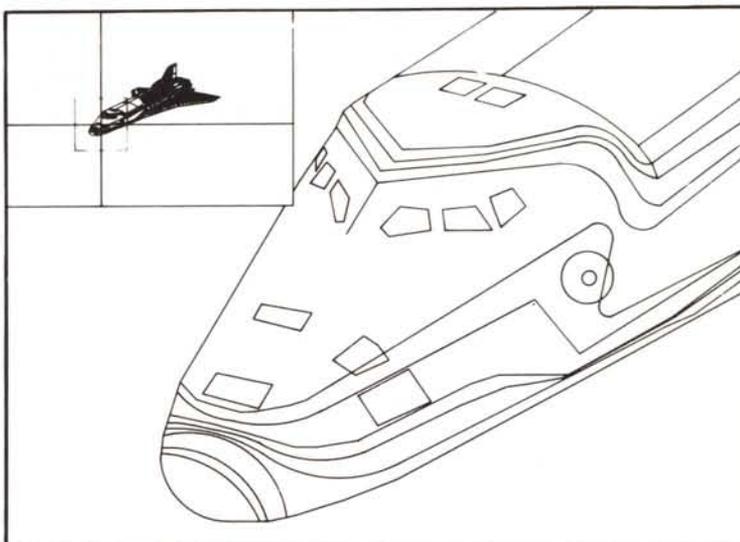
Zoom Pan e Viste  
IMMEDIATI

Il primo Display List  
per la Tua EGA/VGA

Di facile installazione velocizza  
fino a 40 volte le operazioni  
di Zoom e Pan e permette  
l'utilizzo del "BIRDEYE" per  
avere sempre l'intero disegno  
sul monitor.

Supporta le seguenti risoluzioni  
640x350-640x480-800x600-1024x768  
in Singolo o Doppio monitor.

non ha bisogno di Hardware aggiuntivo



**Lightning Zoom N° 1 per AutoCAD**  
**Lightning Zoom in Italiano L. 800.000**  
**Lightning Zoom + VGA1024x768 NI L. 1.500.000**

**SLICK**

- Visualizza e stampa disegni, senza entrare in AutoCAD, direttamente da DOS
- Strutture ed albero delle directory
- PAN e ZOOM sull'immagine

**QUIK SURF**

- Il più veloce generatore di curve di livello per AutoCAD
- Gestisce più di 4000 punti al minuto (386 20 MHz)
- Formato libero per l'input delle coordinate dei punti x-y-z
- Output in file 3D-DXF
- Compatibile con AUTOSHARE ed il comando HIDE
- Generazione di profili
- Calcolo dei volumi

**AUTO PLOT**

- Permette di plottare senza interrompere l'edificazione del disegno

**ADIPRINT**

- Driver per plottere in alta risoluzione su stampante grafica a 9 o 24 aghi portando la risoluzione da 120x72 a 280x240 o 360x180 punti/inc

**CAD FONT**

- 20 font di caratteri utilizzabili in AutoCAD, tra i quali il futura, helvetica, leroy, ecc.

**WORK STATION C.A.D.**

- CPU 80386 SX/16 MHz
- HD 85MB (28ms)
- FLOPPY DISK DRIVE 1.2 MB
- 2 MB RAM
- COPROCESSORE 80387/SX
- 2 PORTE SERIALI
- SCHEDA GRAFICA 1024x768/16 NL.
- MONITOR 16" (20-60 KHz)
- SCHEDA GRAFICA MONOCHROME
- MONITOR 14" MONOCHROME
- DIGITIZER 12" X 12"
- LIGHTNING ZOOM
- SLICK

**L. 9.900.000 .**



**GRAFITEL**

SERVIZI & SISTEMI  
CAD

Viale Liegi,49 - Roma 00198  
Tel.06-863176/8842367

**Leader in prodotti per AutoCAD**

• SISTEMA BASATO SU HARDWARE UNIDATA



• AutoCAD è un marchio Autodesk