Omikron.Basic 3.0 e Omikron.Compiler 2.4

di Massimo Truscelli

Questo mese dedichiamo lo spazio Atari ad un prodotto che viene dalla Germania, terra nella quale i sistemi ST godono senza dubbio di maggior fortuna rispetto a quanto avviene in Italia. Il prodotto in visione è l'ennesima versione del linguaggio Basic, questa volta sviluppato da tali Thomas Kemp e Artur Sodler di chiare origini germaniche; si tratta di una versione del linguaggio di più ampia diffusione che offre una compatibilità molto elevata con l'MBasic Microsoft, nel senso che i programmi scritti con quest'ultimo possono essere facilmente trasferiti in Omikron.Basic, ma non viceversa.
L'importatore italiano, la Hard & Soft di Terni, che rappresenta uno dei pochi distributori italiani di software per Atari, ha curato la traduzione dei manuali dal tedesco in italiano e rende disponibile anche un compilatore che velocizza enormemente l'esecuzione dei programmi realizzati con Omikron.Basic

Descrizione generale

L'Omikron.Basic (si scrive proprio con il punto tra le due parole) offre alcune interessanti caratteristiche come un editor di schermo separato che integra anche tutta una serie di comandi VT52 standard.

La commercializzazione del prodotto avviene su due diversi supporti: uno consiste in una cartuccia hardware da inserire nella Modulport degli Atari, il secondo supporto è invece rappresentato da un classico dischetto da 3.5''.

La versione della quale parleremo in queste note è quella su disco, che rispetto all'altra non presenta sostanziali differenze, al punto che il manuale in dotazione è praticamente identico.

L'installazione è immediata in quanto il boot del sistema può avvenire direttamente dal dischetto e l'Omikron.Basic può essere richiamato semplicemente selezionandolo con il mouse come qualsiasi altro programma.

L'applicazione da lanciare per accedere al Basic è OM-BASIC.PRG, si tratta di un programma che in realtà provvede a sua volta a richiamarne altri, uno in particolare che alloca l'interprete in una zona di memoria protetta in modo che non vada perduto nemmeno premendo il pulsante di reset. Una volta entrati in ambiente Basic lo schermo si presenta con le indicazioni riguardanti la memoria

Omikron.Basic & Compiler

Produttore:
Omikron.Software Erlachstr. 15. D-7534 Birkenfeld 2
Distributore:
Hard & Soft snc. P.zza E. Fermi, 4, 05100 Terni.
Prezzo *(IVA 9% inclusa):*

Omikron.Basic 3.0 in cartuccia	*L. 189.000*
Omikron.Basic 3.0 su disco	*L. 169.000*
Omikron.Compiler 2.4	*L. 130.000*

disponibile, la versione ed i copyright nella forma visibile nel dump dello schermo pubblicato in queste pagine.

L'editor

Abbiamo detto che l'Omikron.Basic offre un editor di schermo separato, ma in realtà la caratteristica principale consiste nel poter disporre di due diversi editor: il primo è quello che appare subito dopo aver richiamato l'interprete ed offre una serie di possibilità piuttosto comuni come l'intervento su linee di programma per l'inserimento di altre linee, la cancellazione di parti di esse o per intero. Con opportune combinazioni dei tasti è possibile disporre di altri comandi riguardanti la cancellazione del contenuto del buffer di tastiera, la creazione di cornici nelle quali saranno visualizzati tutti i messaggi inviati mediante un'istruzione PRINT, la cancellazione dello schermo ed il ritorno allo schermo precedentemente usato in un programma prima del passaggio in modo diretto.

Oltre a questo editor di schermo è possibile attivare anche il Full Screen Editor, quello che in definitiva permette di inserire il «testo» di un programma ed elaborarlo.

Questo editor è organizzato con alcuni menu a discesa che permettono numerose operazioni la più utile delle quali è certamente quella di ricerca e sostituzione.

Le opzioni offerte corrispondono ai seguenti menu a discesa: FILE, FIND, BLOCK, MODE, GO, RUN e riguardano nell'ordine: la gestione dei file su disco (LOAD, SAVE e DIRECTORY); la ricerca e la sostituzione di stringhe all'interno dei programmi (con possibilità di operare anche sui token); la possibilità di intervenire sui blocchi permettendo di selezionarli, salvarli su disco ed eventualmente inserirli dove sia richiesto ogni volta che si desideri.

Altre importanti operazioni sono consentite con il menu MODE: è possibile selezionare il modo inserimento, cioè inserire una riga vuota o cancellarne una ogni volta che è necessario, scegliere la risoluzione video tra diverse configurazioni (25 × 80 caratteri, 44 × 108 caratteri e 57 × 128 caratteri); è possibile suddividere lo schermo in più parti per poter scrivere in un punto e confrontare un altro punto del programma stesso con risoluzioni diverse in base al modo grafico a colori selezionato; si può scegliere di visualizzare i numeri di linea e gli errori ed infine, salvare le condizioni operative prescelte.

I restanti menu permettono di saltare a determinati punti del programma come inizio e fine, oppure a marcatori definiti dall'utente secondo varie modalità e ad eseguire i programmi scritti con l'editor secondo diverse possibilità tra le quali: compilazione ed esecuzione, esecuzione con attivazione del monitor sul funzionamento (TRON), salvataggio ed esecuzione, esecuzione dei file .BAS ed esecuzione dei file .PRG.

La programmazione

L'Omikron.Basic è senza dubbio molto potente e mostra alcune particolarità costitutive molto interessanti.

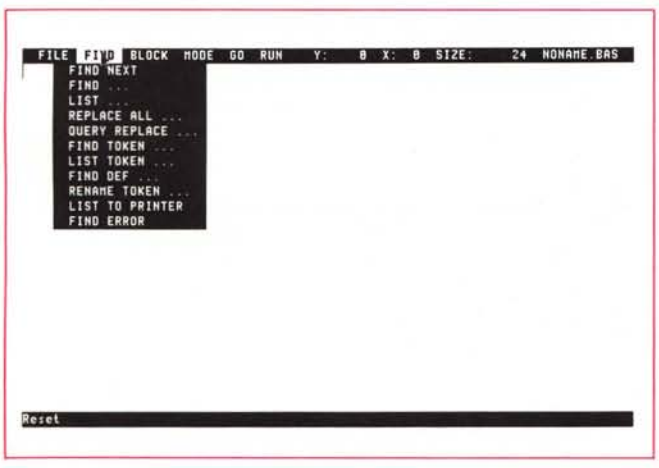
Per cominciare dalle particolarità bisogna innanzitutto dire che tutte le variabili sono automaticamente riconosciute come «long-Integer», ciò vuol dire che se si desidera lavorare con variabili float è necessario inserire l'istruzione DEFSNG «A-Z» come prima riga del programma oppure aggiungere uno specificatore (!) dopo ogni nome di variabile: A!, B!, ecc.

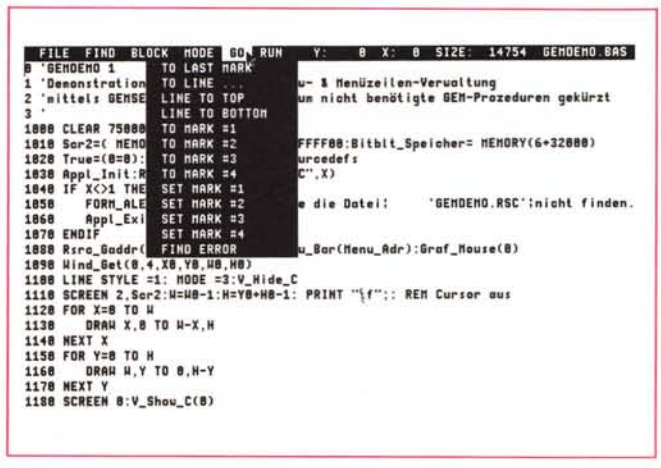
Sempre a proposito di virgola mobile è possibile contare su un vasto range di calcolo offerto da due possibili tipi di variabile float: short-float e long-float. La prima consente un campo di calcolo compreso tra +/− 5.11 * 10^ 4931; la seconda, contraddistinta dal suffisso #, offre un campo praticamente uguale, ma molto più preciso (19 posizioni contro le 9.5 della precedente).

Ciò vuol dire che operando in singola o doppia precisione, si hanno risultati

Il messaggio di copyright dell'Omikron.Basic si presenta in questo modo.

```
***     OMIKRON.BASIC V3.0     © OMIKRON.Software     ***
- Press [Help] to enter editor -
3848694 bytes free.
OK
```





Due dei menu a tendina del Full Screen Editor; si noti la barra inferiore per il Reset.

diversi in termini di fabbisogno di memoria e velocità di elaborazione; specialmente nel trattamento di vettori e matrici: con la precisione doppia si ha un impiego di tempo superiore tra 4 e 8 volte quello della singola precisione.

Gli operatori logici offerti comprendono oltre ai canonici AND, OR, NOT, NAND, NOR e XOR, anche i meno diffusi EQV [il senso del quale corrisponde all'equivalenza di due numeri: ad esempio A=B EQV C, analogo a NOT (B XOR C)]; IMP, dove A=B IMP C determina il significato di implicazione ai numeri B e C in modo bit secondo la solita tabella della verità; SHR e SHL con i quali l'espressione A=B SHR C, oppure A=B SHL C, determinano lo spostamento del numero B a destra (o sinistra) di tante posizioni quante ne indica il numero C, provocando in tal modo una divisione o un raddoppio.

Il vantaggio offerto è notevole in termini di incremento della velocità con i numeri Integer e soprattutto nell'uso del compilatore in confronto ai tradizionali operatori / e *. Oltre alle potenzialità mostrate nel campo delle funzioni matematiche, l'Omikron.Basic mostra di essere dotato anche di funzioni stringa molto articolate. È possibile eseguire operazioni di addizione di stringhe, moltiplicazioni, lettura di stringhe in senso inverso (MIRROR$) e le consute funzioni LEFT$, RIGHT$, MID$, LEN, VAL, STR$, ASC, CHR$, SPC, SPACE$. Oltre alle variabili stringa l'Omikron offre anche la consultazione di alcune variabili riservate che possono dare importanti informazioni sullo stato del sistema; mi limiterò a citare quelle riguardanti la gestione dei movimenti e dei pulsanti del mouse (MOUSEX, MOUSEY, MOUSEBUT), il tempo trascorso dall'accensione del sistema e l'ora attuale nel formato ore:minuti:secondi (TIMER e TIME$), la data attuale, la posizione del cursore e la visualizzazione del contenuto delle variabili riservate alla gestione degli errori (DATE$, CSRLIN, ERR, ERL, ERR$).

Per chi è abituato ad usare delle parti di programma ripetitive che cessano solo quando viene soddisfatta una determinata condizione, i cosiddetti cicli, è possibile contare sui costrutti: REPEAT...UNTIL; WHILE...WEND ed il più conosciuto FOR...NEXT.

Molto utili sono anche le istruzioni riguardanti i vettori, le funzioni e la definizione di procedure; SORT permette di ordinare vettori monodimensionali secondo un valore crescente. Le funzioni create con Omikron.Basic possono essere di due tipi, monolinea e multilinea e si differenziano nel tipo di definizione e soprattutto nel campo di possibilità che offrono. In particolare, le funzioni multilinea offrono più possibilità legate al confronto di parametri senza dover necessariamente ricorrere alla condizione logica IF...THEN e possono essere anche ricorsive come nel caso del calcolo del fattoriale di un numero.

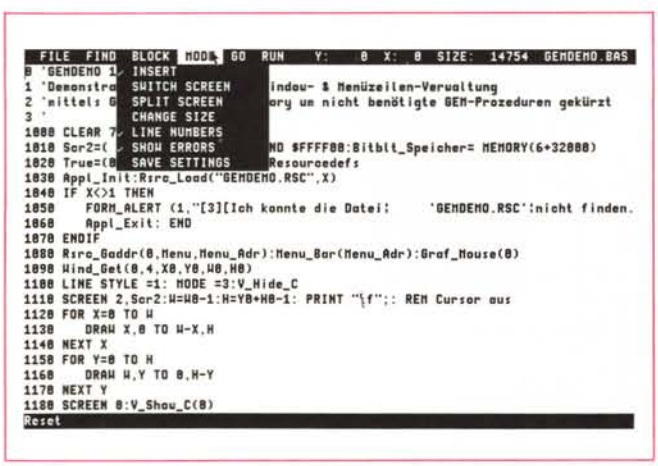
Per ciò che riguarda la definizione di procedure basta dire che è possibile definire dei comandi a piacimento semplicemente usando opportunamente l'istruzione DEF PROC. Una caratteristica importante degli Atari ST è quella di essere previsti per poter funzionare in multitasking. In realtà, sebbene con multitasking si inteda l'esecuzione contemporanea di più applicazioni, il sistema ST, come tutti i sistemi multitasking, esegue un programma dopo l'altro alternando un pezzo dell'uno e dell'altro; per l'utente finale il risultato è tale da far credere che i programmi vengano eseguiti nello stesso momento; logicamente le capacità dell'ST non sono particolarmente elevate, ma per l'utente può essere utile disporre anche di queste limitate capacità specialmente nell'uso di programmi per impiego professionale. Omikron.Basic dispone di una serie di istruzioni riservate alla gestione multitasking delle risorse dei sistemi ST Atari. I comandi di multitasking sono: ON KEY GOSUB, ON MOUSEBUT GOSUB, ON HELP GOSUB, ON TIMER GOSUB. Nell'ordine provvedono a saltare ai sottoprogrammi corrispondenti non appena viene premuto un tasto, si agisce su un pulsante del mouse, si preme il tasto HELP, oppure ogni volta che sia trascorsa una certa quantità di secondi precedentemente indicata.

Tutte le istruzioni multitasking offerte da Omikron.Basic presentano alcune caratteristiche comuni: il controllo continuo di un evento particolare nel corso dell'esecuzione normale del programma; ad evento verificatosi, il controllo del programma salta ad un sottoprogramma legato all'attuazione dell'evento ed una volta avvenuta l'esecuzione del sottoprogramma il ritorno del controllo al programma principale nel punto nel quale si era interrotto dal verificarsi dell'evento.

La descrizione di tutte le altre caratteristiche di Omikron.Basic potrebbe continuare ancora per molto, ma considerando lo spazio a disposizione forse è il caso di dare un'occhiata anche al compilatore che amplia le già ottime prestazioni di Omikron.Basic.

Omikron.Compiler

Per chi non lo sapesse (credo sinceramente molto pochi), nell'esecuzione di un programma, la CPU deve innanzitut-

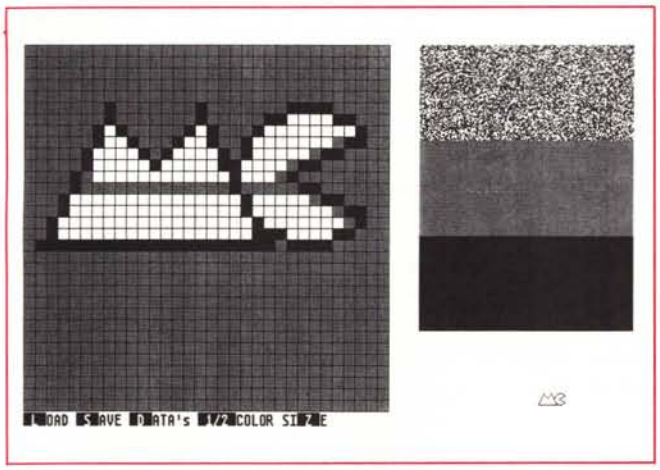


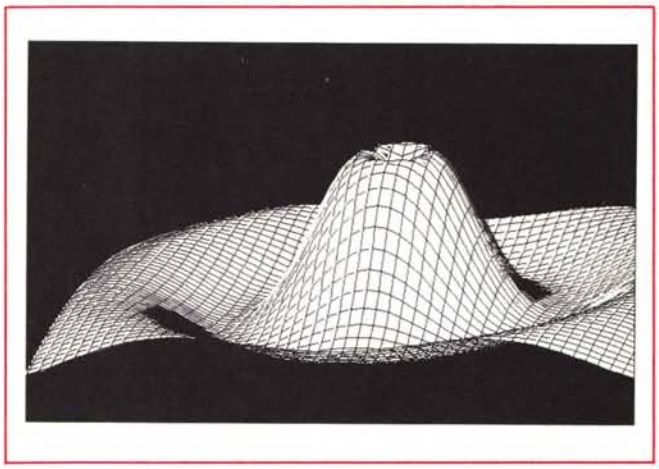
Due diverse risoluzioni settate dal sottomenu MODE: a destra è visibile la seconda finestra settata dall'opzione SPLIT SCREEN per la correzione del codice sorgente dagli errori visualizzati nella prima.

```
Serial # 6819              OMIKRON.BASIC Compiler V2.4

Compiling data...

Warning: Unused statement(s):
PROC MENU_BAR(?); PROC GRAF_HANDLE(?,?,?,?,?); PROC GRAF_MOUSE(?,?,?,?,?); PROC
FSEL_INPUT(?,?,?); PROC WIND_GET(?,?,?); PROC VSIM_MODE(?,?); PROC V_SHOW_C;
Compiling line...
# 64922
# 64922
Bytes free: 575246 BASIC
            3219488 System

Bytes used: 13828 BASIC
            13744 System

Compilation time: 8.385 sec.
```

Un esempio di ciò che il compilatore visualizza quando non riconosce alcune istruzioni. Sul manuale sono chiaramente indicate quali sono le istruzioni non riconosciute e come ovviare all'inconveniente.

Omikron.Basic possono non funzionare con il compilatore, caso nel quale apparirà un chiaro messaggio di avvertimento sullo schermo in fase di compilazione. In queste evenienze il manualetto in dotazione consiglia di adattare le istruzioni che possono creare problemi alla struttura del compilatore suggerendo





Due applicazioni sviluppate con l'uso dei prodotti Omikron: uno Sprite Editor ed il grafico di una funzione 3D.

to decodificare le istruzioni del linguaggio utilizzato (Basic, Pascal, ecc.) in codici direttamente eseguibili. Questa operazione viene svolta dall'interprete e costa un po' di tempo ogni volta che il programma viene avviato. Il linguaggio macchina (L.M.) rappresenta quel codice direttamente eseguibile dal processore e la funzione di un compilatore, per chi non avesse ancora capito, è proprio quella di tradurre le istruzioni di un linguaggio di programmazione in un codice L.M. senza dover passare per l'interprete.

L'Omikron.Compiler è stato realizzato proprio per permettere la compilazione dei programmi scritti in Omikron.Basic per il raggiungimento di maggiori velocità di esecuzione.

Sul dischetto del Compiler, tra l'altro realizzato con lo stesso Omikron.Basic e successivamente compilato da... se stesso, sono presenti alcuni programmi dimostrativi di un certo effetto in termini di prestazioni in velocità.

Il programma che più di tutti mostra le differenze tra la versione interpretata e quella compilata è il solito programma per la visualizzazione sullo schermo con effetto tridimensionale di una funzione matematica, del quale vedete riprodotto l'output. I tempi di realizzazione della figura sono notevolmente diversi tra le due versioni e mostrano effettivamente come il programma compilato risulti essere molto più veloce rispetto alla versione Basic.

Il funzionamento del compilatore è immediato in quanto non necessità di alcuna particolare precauzione. Si effettua la scelta del programma da compilare con l'apposito Item Selector e quindi si avvia la compilazione confermando la scelta.

La compilazione avviene molto velocemente e nel corso di essa è possibile seguire sul monitor alcune indicazioni riguardanti i numeri di linea letti e quelli già compilati oltre alle indicazioni sulla memoria utilizzata dal sistema e dal Basic. A compilazione eseguita si può vedere sullo schermo anche il valore corrispondente al tempo utilizzato per la compilazione.

Per avere un'idea delle prestazioni del compilatore basta dire che un programma della lunghezza di circa 48 Kbyte viene compilato in poco più di un minuto. Una precauzione da prendere è quella di accertarsi che sul dischetto contenente il codice sorgente ci sia abbastanza spazio per ospitare anche il compilato e soprattutto che il programma compilato sia su un dischetto nel quale sia presente anche il file BASLIB, pena la non esecuzione del programma.

Alcuni programmi perfettamente funzionanti con la versione interprete di quali sono i problemi più facili da riscontrare e le relative vie per la loro risoluzione.

Conclusioni

In definitiva si tratta di due ottimi pacchetti destinati agli sviluppatori di software professionisti, ma anche a chi è interessato a scriversi i propri software solo per diletto. Il prezzo è sicuramente molto conveniente.

La qualità è buona specialmente se si tiene conto di alcune caratteristiche; un plauso va all'importatore per la buona traduzione dei manuali che spiegano in maniera abbastanza dettagliata tutte le caratteristiche del prodotto e soprattutto hanno inserite alcune indicazioni di carattere generale nell'architettura dei sistemi ST e sulle loro caratteristiche hardware; tutte informazioni che possono risultare molto utili sia agli sviluppatori professionisti di software che agli «smanettoni».

Specialmente usando il compilatore si ottengono prestazioni paragonabili a quelle ottenibili adoperando altri linguaggi di programmazione più potenti.

Certo, per una macchina equipaggiata con un processore 68000 sarebbe forse il caso di pensare ad usare un linguaggio più impegnativo... e, almeno, a non limitarsi al buon vecchio Basic. MC