



Questa newsletter italiana nasce da un accordo avuto con Chris Schneider che ha acconsentito alla traduzione italiana della sua *Newsletter Shift838*. Vuole essere una nuova risorsa per i computer **TI-99/4A** e **GENEVE 9640**, in modo da divulgare e mantenere aggiornati tutti gli utenti che ancora oggi si interessano a questi computer.

*“Per prima cosa voglio ringraziare Chris Schneider (fondatore ed editore di SHIFT838) per avermi permesso di aiutarlo con il numero di questo mese. Il suo bollettino è stato una boccata di ossigeno apprezzata da tutti gli appassionati del TI-99/4A e ho colto al volo l'opportunità di poter offrire il mio aiuto come ospite-editore per il mese di maggio. Per questo mese, ho scelto di focalizzare un po' d'energia e d'attenzione verso il Forth sul TI-99/4A. Credo che sia uno dei più potenti linguaggi, anche se sottoutilizzato, disponibili agli sviluppatori di giochi e di applicazioni per il sistema e, auspicabilmente, sentirne parlare da parte degli esperti e vedere se qualche codice Forth incoraggerà alcuni di voi a offrire una prova del Forth.”*

**--Owen Brand, "Opry99er" --**

In questa newsletter si parlerà con **Lee Stuart** e **Mark Wills** (sono due Guru per il linguaggio Forth) e ci racconteranno della loro esperienza e considerazioni riguardo questo linguaggio, approfondiremo anche questi argomenti:

- **Un modulo chiamato Fourth in fase di costruzione;**
- **Esempio (con descrizione del listato) di programmazione Forth;**
- **Il gioco “Dark Star” programmato da Mark Wills;**

Buona Lettura !

[Ciro Barile](#)

# Interviste

## Che cosa è il Forth?

Forth è uno stack-basic, cioè un linguaggio di programmazione LIFO (Last-In-First-Out) Reverse-Polish-Notation che “enfatisza un dizionario estendibile su sintassi”.

## Che cosa vuol dire tutto questo?

Ho pensato di porre a due dei guru della nostra comunità TI (*Lee Stewart* e *Mark Wills*) qualche domanda sulla loro lunga esperienza con il Forth, come sono diventati autori di linguaggi Forth impostati su un singolo modulo e perché (con le loro stesse parole) il Forth dovrebbe essere preso in considerazione dai programmatori software per i nuovi giochi e le nuove applicazioni sul TI da ora in poi. La nostra conversazione segue ...

**D: Grazie per aver accettato di parlare con me qui, ragazzi! Mi piacerebbe iniziare chiedendo a ciascuno di voi di spiegare brevemente le vostre rispettive release Forth per il TI99 ed i vostri obiettivi individuali, in via preliminare, dei vostri progetti.**

**Lee:** *fbForth* si è evoluto da un progetto che ho iniziato non molto tempo dopo aver avuto tra le mani il TI Forth alla fine del 1983 o all'inizio del 1984, progetto che era quello di convertire il manuale TI Forth, che avevo ricevuto in WordPerfect dal DC User Group, in una versione per il TI Writer. Volevo riformattarlo e modificarlo per renderlo più facile da usare. Quel progetto ha languito fino a circa quattro anni fa, quando ho trovato tutte le attività del TI-99/4A sul forum di AtariAge, cui ho subito aderito. Ho convertito il manuale in formato OpenOffice Writer e ho prodotto una migliore edizione in PDF circa nove mesi più tardi.

Mentre completavo una seconda edizione, un anno dopo ho cominciato a fremere per convertire il TI Forth da sector-based I/O dei blocchi Forth ad una file-base. Il file I/O a blocchi acquista più senso perché in tal modo utilizzare dischi di dimensioni diverse è più facile ed è molto più sicuro del sector I/O che può distruggere un disco se non si fa attenzione. Il *TurboForth* di *Mark Wills* è stato la mia ispirazione principale per la conversione. Ho finito il file-based TI Forth, che ho rinominato fbForth e l'ho rilasciato come *fbForth 1.0* (cioè "file-based Forth") nel mese di novembre del 2013. Ha fatto un paio di miglioramenti, tra i quali c'erano un text mode a 80 colonne, un editor avanzato e una routine per i joystick che può utilizzare il CRU come il TurboForth o il KSCAN come TI Forth.

Quasi subito dopo l'uscita di fbForth 1.0, ho iniziato a lavorare su una versione su modulo, che è culminata con la sua uscita al Chicago Faire nel novembre 2014. E' molto migliorata rispetto alla versione precedente. L'editor 40/80 colonne è stato riscritto in linguaggio assembly (ALC) così come lo erano tutte le grafiche originali, il che ha aumentato notevolmente la loro velocità. La riscrittura dell'editor è stato fortemente influenzato dall'editor presente nel TurboForth. Ho incluso anche un nuovo editor di font. Era anche inclusa una mia implementazione della biblioteca in virgola mobile per l'MDOS del Geneve.

**Mark:** *TurboForth* è il risultato di una riflessione per progettare e costruire un computer single-board. Ho deciso di utilizzare un TMS9995 (stessa CPU come Geneve) con una semplice mappa di memoria e una porta seriale. Avere una porta seriale avrebbe voluto dire che non avrei dovuto implementare l'hardware per una tastiera o un display.

Ho comprato i componenti in un impeto di entusiasmo e ho iniziato a pensare su che tipo di sistema operativo avrei adottato per questo meraviglioso piccolo computer. Dopo aver esaminato diverse opzioni, ho finalmente inciampato, o dovrei dire, ri-imbattuto nel Forth.

Ero stato introdotto nel Forth nel 1989 da Stephen Shaw (una leggenda del TI99 del Regno Unito). Mi fece una dimostrazione del linguaggio su un TI (TI Forth). Ho pensato che fosse fuori di testa. Reverse Polish Notation? Chi diavolo vuole scrivere con una roba così primitiva? Stacks? Puah! Allora respinsi il suo consiglio (nel 1989 avevo 18 anni) e non ho mai più ripensato al Forth, fino a quando non ho visto listati in TI Forth pubblicati su TI\*MES (una rivista inglese del TI-99/4A), dopo di che mi misi a ridere indignato per quanto fossi stato tanto stupido. Oh, cosa vuol dire essere giovani e molto stupidi!

Un veloce excursus al 2008/2009 (se non ricordo male). Mi sono imbattuto in alcuni articoli su internet su come sia stato semplice implementare un sistema con il Forth su hardware minimali, e non solo, ma il linguaggio di programmazione era il sistema operativo e il sistema operativo era il linguaggio di programmazione. Nessun confine. Nessuna barriera. Anche la maggior parte del Forth stesso potrebbe essere scritto in Forth. Questo è stato sufficiente per trascinarci, agganciarci, metterci in riga e farci crollare!

In fretta abbandonai l'idea di costruire un piccolo computer. I chip sono ancora fermi in un cassetto da qualche parte. Io invece decisi di scrivere un sistema Forth per il TI-99/4A, il che mi portò ad un viaggio dove feci una scoperta che culminò nel TurboForth. Il TurboForth è un ampio progetto. Migliaia di righe di codice. Molte

parti di esso sono state riscritte quando ho scoperto di più sul Forth (più di un sintomo della mia incapacità di fare ricerche adeguate che di qualsiasi altra cosa!), fu sviluppato a singhiozzo, come sono la maggior parte dei progetti hobbistici.

E' un progetto veramente internazionale; grandi parti sono state scritte ad Aberdeen, in Scozia. Ricordo di aver scritto l'editor in circa 5 ore in una giornata di sole nel mio ufficio (mi sono preso una pausa di lavoro!). Il file di I/O è stato scritto all'International Hotel a Lang Fang, Cina. Le istruzioni per le 32 colonne (DCHAR, HCHAR, COLOR, SPRITE e altre) sono state scritte a Tashkent in Uzbekistan. Altre parti sono state scritte a Bhukara in Uzbekistan.

I miei obiettivi per il TurboForth erano semplici: ho voluto che fosse veloce così come avrei voluto farlo. Ampie parti di esso sono scritte in linguaggio assembly e un sacco di esso sono altamente ottimizzate - essendo state scritte e riscritte diverse volte per salvare un'istruzione qua e un'istruzione là. Ho anche invertito la direzione degli stack lunghi dopo che lo avevo (presumibilmente) "finito" (anche se un progetto di software quando mai può dirsi finito?) perché un codice più breve è più veloce. Più veloce è buono e l'essere più corto mi ha permesso di inserire più codice nei 16K di ROM!

**D: In gamba, ragazzi!**

**Sembra che ci siano state ragioni differenti per iniziare i vostri progetti, ma i vostri progetti molto simili si sono incontrati ... Come è successo che poi vi siete conosciuti e quale influenza (se c'è stata) è avvenuta sui vostri rispettivi progetti?**

**Mark:** Se ben ricordo, i miei primi colloqui con Lee avevano a che fare con un programma per rintracciare un Social Security Number unico nel suo genere che Lee aveva scritto in TI Forth e poi convertito al TurboForth. IIRC (*se ricordo bene ndt*), ci sono stati alcuni problemi che sono emersi durante il porting del codice. Nel frattempo Lee aveva scoperto che la versione V1.0 del TurboForth non attuava la floored integer division, o lo faceva, ma in modo non corretto, e questo creava vari problemi. Questo ha creato uno scambio messaggi gigantesco (tutto tramite AtariAge) di come la floored integer division lavorava sul Forth-83 standard che io stavo seguendo. Stavo scrivendo il codice e stavo per inviare una versione del TurboForth a Lee da provare! Ci sono voluti un paio di tentativi, ma ce l'ho fatta alla fine!

Ci sono state parecchie collaborazioni lungo la strada. Per esempio, quando ho sviluppato l'assemblatore per il TurboForth (un port dell'assembler del TI Forth) Lee mi ha assistito leggendo la bozza della documentazione. Lee mi ha anche aiutato con la documentazione della libreria a 32 bit. Inoltre, Lee ha contribuito anche con un'intera biblioteca in virgola mobile (completa di un'ampia documentazione) per il

TurboForth. Io, a mia volta, ho aiutato Lee con recensioni dei manuali fbForth e la segnalazione di bug occasionali. E' stato davvero bello vedere l'fbForth giungere a buon fine. Lee lavora molto più veloce di me!

**Lee:** Se io lavoro più veloce di te, Mark (e comunque ne dubito...), è perché io ho molto più tempo libero disponibile ☺. Sono andato in pensione 6 anni fa.

**D:** Lee, so che mantenere una perfetta compatibilità con gli standard dell'originale TI Forth è stato importante per te. È stata una limitazione a tutto il tuo lavoro? Sicuramente ad un certo punto hai avuto un paio di decisioni da prendere sulla questione se "allontanarsi o non allontanarsi" dall'origine ?

**Lee:** Mantenere una perfetta compatibilità con gli standard del TI Forth originale era sicuramente importante come obiettivo; ma, fin dall'inizio, è stato impossibile a causa del cambiamento dall'accesso al direct-sector per l'I/O dei blocchi (schermi, nella terminologia figForth) all'accesso ai file.

Per quanto riguarda l'"allontanarsi o no", ho cercato di non farlo a meno di non avere un'altra scelta. Non avere una scelta coinvolge di solito il nuovo file-based I/O per i blocchi. L'appendice E in ciascuno dei due manuali per il fbForth 1.0 (versione disco) e per il fbForth 2.0 (versione per modulo, nonché con vari miglioramenti) dettaglia come rimuovere, aggiungere e modificare le words. Se mi sono comunque allontanato dal codice originale in qualche punto, è stato probabilmente nell'aggiunta di words al dizionario residente per migliorare il fbForth. Nel fbForth 2.0, ho caricato la maggior parte delle words opzionali sul dizionario residente nello spazio ROM perché l'ho potuto fare. Questo ha drammaticamente accelerato le words per i grafici, alcuni dei quali erano così lenti da essere praticamente inutilizzabili, soprattutto LINE in modalità bitmap. L'unico svantaggio di questa modifica è che ciò ha causato un maggior aumento di tempo per l'interprete nel compilare il dizionario per le words poste nella parte superiore (prime voci) del dizionario-sul dizionario vengono ricercate iniziando da una lista che parte dalla word definita più recente. Questo, naturalmente, incide negativamente sul tempo di caricamento dei blocchi del disco.

**D:** Ci sono stati alcuni sviluppi incredibili nella comunità di TI ultimamente ... Vale a dire una quantità folle di ROM e GROM. Cosa ne pensi circa l'ampliare i linguaggi in modo da includere più word o funzioni nello spazio del modulo?

**Lee:** La versione modulo dell'fbForth (v2.0) è molto più estesa rispetto alla v1.0. Ho trovato un modo per dividere le dictionary header del dizionario dalla main resident dictionary bank; ma vi è troppo poco spazio in una di quei bank per

un'ulteriore espansione.

Più bank avrebbero consentito una maggiore ALC nella ROM; ma avrei dovuto pensare in modo intelligente su come consentire i dictionary body e gli header di essere presenti in più di un unico bank prima che lo spazio aggiunto fosse usato per aggiungere altre words.

Una cosa da tenere a mente quando si aggiungono words al dizionario è che aumentando la dimensione del dizionario aumenta il tempo necessario per caricare il codice sorgente dell'interprete dai blocchi. Inoltre aumenta il tempo dal terminale; ma, non è quasi percettibile.

**Mark:** Sì. Il TurboForth vive in una EPROM 16K. Sono abbastanza colpito dal fatto di averlo potuto comprimere così tanto. Ho fatto una certa quantità, un po' folle, di ottimizzazione – rivisitando il codice più e più volte – rimescolando varie cose. Bob Carmany, che ha fatto la maggior parte dei test del TurboForth (a partire nel lontano 2009!), è stato costantemente subissato dalle mie e-mail dicendo "Ehi Bob, lo sai che non abbiamo il posto per metterci le nuove funzioni? Lo dobbiamo creare ora!" - Ma si sono raggiunti dei limiti invalicabili, ora è pieno zeppo ☺ .

**D:** Un'ultima domanda per voi entrambi.

**Nella nostra comunità (che sembra crescere in modo abbastanza costante in questi giorni) ci sono molte opzioni per l'aspirante sviluppatore ...**

**dal vostro punto di vista, perché un programmatore dovrebbe dare un'occhiata al Forth per sviluppare giochi e applicazioni, diciamo, sorvolando sull'Extended Basic o l'Assembly?**

**Mark:** Perché risulta più soddisfacente e per l'efficienza della memoria. I Forth, sia fbForth che TurboForth, sono più di 10 volte, e in alcuni casi, centinaia di volte più veloce del TI BASIC o dell'Extended Basic.

Inoltre, vi aprono completamente tutta la macchina. Nulla vi è nascosto, come in TI BASIC o in Extended Basic. Inoltre, non ci sono sequenze tipo scrivi/compila/carica/ripeti la battuta come quando si scrive in Assembler o in C. Basta digitare il codice e provare, proprio come in BASIC. C'è sicuramente una curva di apprendimento quando si inizia a studiare il Forth. Non è un linguaggio facile da imparare. Diventa ancora più difficile se si è abituati a programmazione in BASIC, perché il Forth è molto diverso. E' come avere un bambino! Trasforma completamente il tuo mondo per un po' e non si riesce a capire cosa diavolo stia facendo, o richiede una maggiore quantità di tempo. Ma alla fine si arriva a conoscere l'altro, e le ricompense sono enormi rimanendoti per tutta la vita.

**Lee:** Concordo! Per citare un pensiero sul Forth di *Leo Brodie*, "Forth è un linguaggio, un sistema operativo, un insieme di strumenti, e una filosofia. Si tratta di un mezzo ideale per pensare perché corrisponde al modo in cui funziona la nostra mente."

Penso che il più grande ostacolo per l'apprendimento del Forth è la sua natura di stack-oriented. Una volta che l'idea ti entra dentro, non si può più tornare indietro.

## **Risorse per il TurboForth & l'fbForth**

Forum del TI-99/4A su AtariAge:

<http://ti99.atariage.com>

Principale guida all'fbForth con il rilascio continuo di aggiornamenti al primo post:

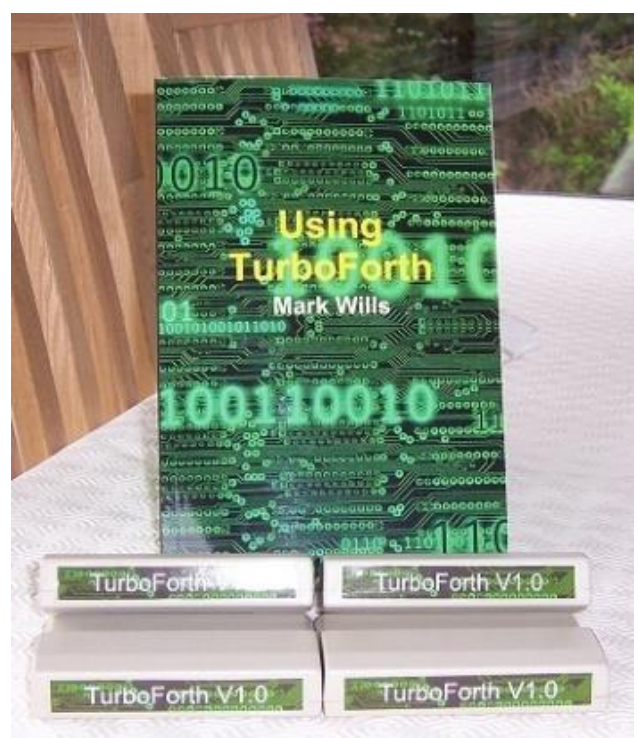
<http://atariage.com/forums/topic/210660-fbforthti-forth-with-file-based-block-io-post-1-updated-12052014/>

Principale risorsa per il TurboForth:

<http://www.turboforth.net>

Per entrambi ci sono risorse qui:

<http://atariage.com/forums/topic/153704-ti-994a-development-resources/>

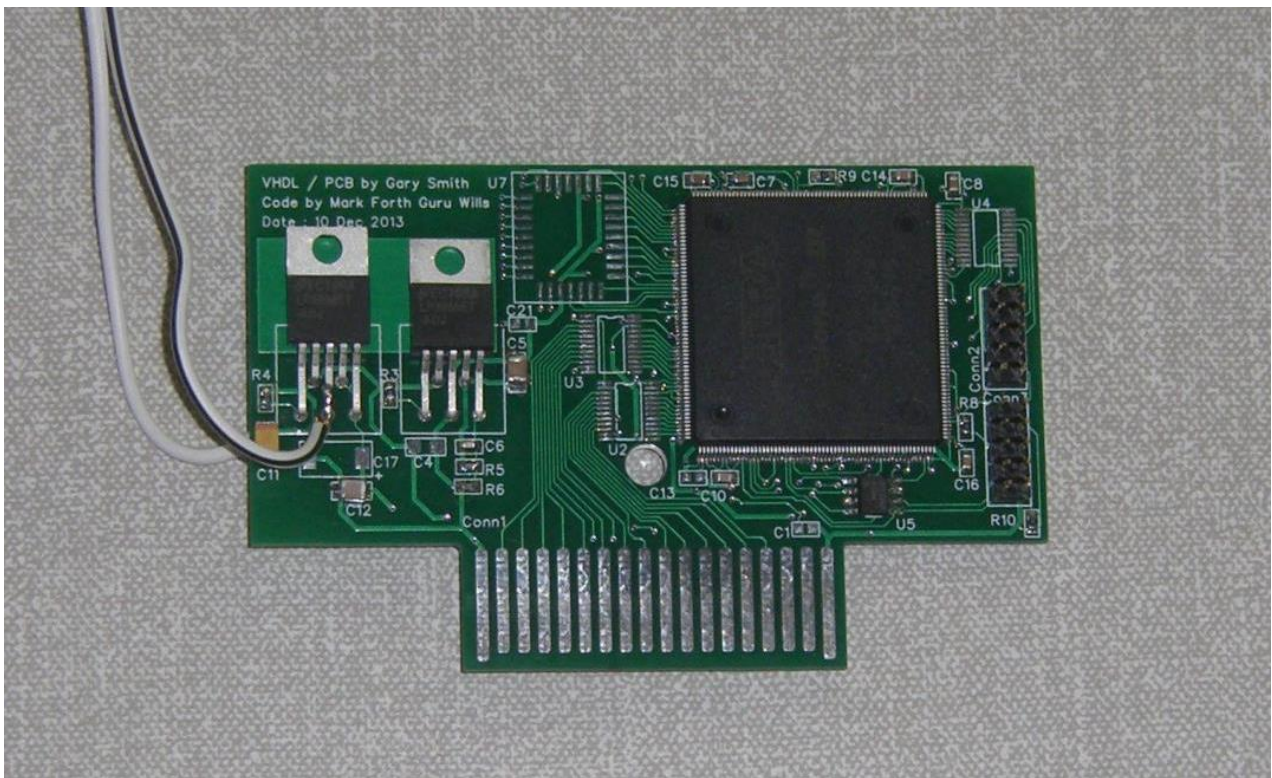


# Hardware

Nel mondo del Forth sul TI-99/4a, nulla per il futuro sembra più intrigante di un progetto di Mark Wills e Gary Smith, a volte indicato come modulo "**Fourth**" (non FORTH). Si tratta di un modulo basato su un FPGA che esegue il Forth con 4 (four!) processori separati, ciascuno in grado di eseguire 3 istruzioni per ciclo di clock.

Questi processori utilizzano il Forth come loro linguaggio macchina (..davvero molto affascinante). Sarà abbinato con un'interfaccia unica Forth sul lato del TI, che somiglia al TurboForth, ma non sarà derivata direttamente dall'architettura TurboForth già esistente. La scheda prototipo PCB è completa e nelle mani degli sviluppatori.

Si parla di aggiungere una scheda SD di lettura/scrittura che consente al TI di lavorare in un ambiente operativo del tipo DOS. Le possibilità con questo progetto sono praticamente infinite, e io, per primo, non vedo l'ora di vedere questa "cosa" collegata al mio sistema TI-99/4A.





# Coding

In linea con il tema Forth, Lee Stewart ed io abbiamo lavorato insieme su un piccolo progetto di conversione. Ho voluto mettere insieme un confronto XB/Forth, così ho frettolosamente scritto un pezzetto di codice XB, e Lee mi ha aiutato con la versione Forth. Faccio notare che fbForth è maledettamente più veloce dell'XB e si è dovuto rallentare l'esecuzione tramite le istruzioni "DELAY".

## XB DEMO (TI Extended Basic)

```
100 C = 1
110 CALL CLEAR
120 CALL SCREEN(9)
130 CALL CHAR(42,"00E0FFD5FFFFE74200007F55FF7F7722007052727FFF7F32") 140 DISPLAY AT (6,
C+1):"*+++",
150 DISPLAY AT (6, C):" ";
160 IF C=12 THEN GOTO 180
170 C=C+1 :: GOTO 140
180 CALL SOUND(75,1397,1):: CALL SOUND(75,220,1):: CALL SOUND (75,1397,1) 190 DISPLAY AT
(12,4):"RIDE THE FORTH TRAIN!!"
200 GOTO 200
```

## FORTH DEMO (fbForth 2,0 di Lee Stewart):

```
: DELAY 1280 0 DO LOOP;

: TADA BEEP DELAY HONK DELAY BEEP
;
HEX
: GRAPHSET
  GRAPHICS 9 SCREEN
  COLTAB 20 19 VFILL
  00E0 FFD5 FFFF E742 2A CHAR
  0000 7F55 FF7F 7722 2B CHAR
  0070 5272 7FFF 7F32 2C CHAR
;
DECIMAL
: MOTION
  12 0 DO
    I 6 GOTOXY ." * +++, "DELAY
  LOOP
;
: BANNER
  4 12 GOTOXY ." RIDE THE FORTH TRAIN!!" CR
;
: WAIT
  BEGIN ?TERMINAL UNTIL
  TEXT ;
: FTRAIN GRAPHSET MOTION BANNER TADA WAIT ;
```

## Spiegazione del Programma:

Entrambi questi listati sono "tipo-in" ready. Per il programma fbForth, è sufficiente aprire un nuovo blocco dell' fbForth 2.0, digitare il codice e spedirlo su disco. Poi si carica il blocco.

Questo è un programma molto semplice ... Cambia il colore dello schermo in rosso, visualizza un trenino, lo mette al centro dello schermo, una colonna alla volta, poi visualizza un banner con un piccolo effetto sonoro.

```
BLOCK #1
0 0 1 2 3
0 : DELAY 1280 0 DO LOOP ;
1 : TADA
2 BEEP DELAY HONK DELAY BEEP
3 ;
4 HEX
5 : GRAPHSET
6 GRAPHICS 9 SCREEN
7 COLTAB 20 19 VFILL
8 00E0 FF05 FFFF E742 2A CHAR
9 0000 7F55 FF7F 7722 2B CHAR
10 0070 5272 7FFF 7F32 2C CHAR
11 ;
12 DECIMAL
13
14
15
F1:Del F2:Ins F3:Del Line F4:Nxt Block
F5:30 spcs L/R F6:Prv Block F7:Del EDL
F8:Ins Line F9:Exit ESDX:Cursor ^V:Tab
ENT:Nxt Line ^8:Ins Blank Line FV:-Tab
```



*(Nelle foto sopra ci sono degli screenshot del BLOCK #1 di 2 del programma 'FTRAIN' nel blocco editor dell'fbForth 2.0. La seconda foto a destra è invece il programma eseguito in fbForth)*

Questi programmi sono eseguiti virtualmente alla stessa velocità, grazie all'istruzione DELAY, definito da noi nell'fbForth 2.0 come un ciclo vuoto che viene eseguito 1280 volte prima di terminare.

In Forth, poche word sono definite (come il ritardo DELAY di cui sopra) e poi combinate in una definizione più ampia per eseguire una serie di parole (come la definizione FTRAIN) ... Una volta che tutto questo è fatto, è sufficiente digitare la parola che si trova al più alto livello nel blocco per eseguire il programma. In questo caso, abbiamo definito 6 parole uniche e una parola che li unisce. Diamo un'occhiata a queste definizioni di parole una alla volta:

**DELAY:** Crea un ciclo di ritardo (istruzione necessaria per rallentare l'esecuzione dell'fbForth giù fino a compararlo all'XB)

**TADA:** Riproduce l'effetto audio (fa uso anche della word DELAY nella sua definizione)

**GRAPHSET:** Imposta la grafica per il treno (i colori e le definizioni CHAR)

**MOTION:** Motion è un ciclo che permette il movimento del treno sullo schermo (usa anche DELAY nella sua definizione)

**BANNER:** Visualizza il messaggio sotto il treno

**WAIT:** Simile al comando XB nel codice XB (>200 GOTO 200)

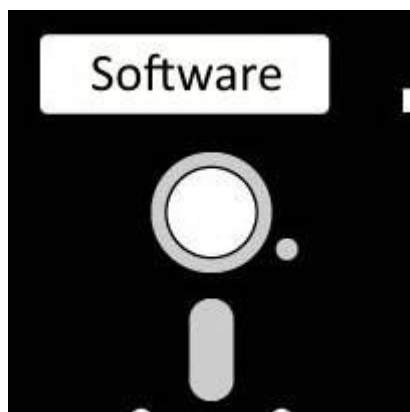
**FTRAIN:** il massimo livello della word ... unisce tutte le parole insieme in una sorta di "filastrocca".

Una delle cose che preferisco di FORTH è la sua capacità di leggere le definizioni poste al massimo livello come frasi. Si può leggere da sinistra a destra e "vedere" nella tua intestazione quello che dovrebbe fare. In questo caso:

```
: FTRAIN GRAPHSET MOTION BANNER TADA WAIT;
```

Si notino i due punti all'inizio di ogni definizione e il punto e virgola alla fine. Queste sono words, proprio come DELAY o MOTION. I ":" è una word che dice al compilatore "quello che segue è una definizione" e ";" è una word che dice al compilatore "quello che mi precede (fino ai ":" ) fa parte di una definizione. Ogni parola deve essere separata da uno spazio in modo che il compilatore possa capire che cosa la vostra definizione stia facendo.

Questo può essere una difficoltà per i nuovi programmatori Forth. DELAY offre veramente all'utente la libertà di "parlare" al computer, ed entrambi fbForth e TurboForth includono le biblioteche iperveloci e tonnellate di word incorporate. Suggesto a tutti la lettura di "GIVE FORTH A TRY (Cimentarsi con il Forth)" sia per l'fbForth che per il TurboForth. Potreste semplicemente innamorarvi.



## Software

Dal momento che abbiamo discusso fbForth e TurboForth in dettaglio, diamo uno sguardo a un programma scritto da Mark Wills in TurboForth chiamato "[Dark Star](#)".

Questo gioco è estremamente divertente, stimolante, e ha un vero e proprio fattore di replay! Mi piace anche il fatto che il codice sorgente del gioco sia completamente descritto sul sito web [www.turboforth.net](http://www.turboforth.net).

Per quelli di voi che stanno prendendo in considerazione "Thinking Forth", questo codice sorgente potrebbe facilmente essere il "Tombstone City" di TurboForth. Il gioco è gratuito per il [download](#), e c'è un video su [YouTube](#) del gioco in movimento. Veramente uno dei miei scacciapensieri preferiti sul TI-99 comunque scritto in qualsiasi linguaggio.

### Dall'autore (Mark Wills):

Dark Star è un classico puzzle game. È necessario raccogliere tutti i "cristalli del potere" in ciascuno dei 25 livelli, nella lotta contro il tempo. Saremo al controllo di un robot che sembra un po' come una biglia che rotola in spiaggia(!).

Il robot può muoversi su, giù, sinistra e destra. Quando si sposta, continua a muoversi fino a quando non colpisce un ostacolo (ad esempio il muro).

Non tutti i cristalli del potere sono raggiungibili direttamente; si dovrà utilizzare un secondo robot come 'bloccante' temporaneo per dare al primo robot la possibilità di fermarsi e creare le giuste traiettorie.

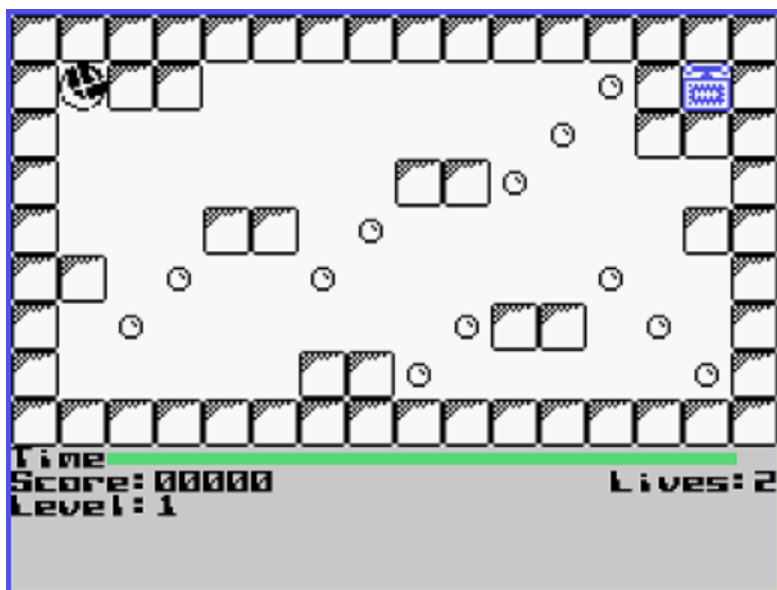
Il gioco offre una dimostrazione lampante di alcune delle funzionalità del TurboForth, di:

- *Animazione Grafica;*
- *Voce;*
- *Suono;*
- *Funzionamento della tastiera o del joystick;*
- *Scrolling dello schermo;*

Vengono utilizzate un certo numero di tecniche di programmazione Forth:

- Esecuzione vettoriale;
- Creazione delle words;
- Valori;

Questo gioco è basato sulla versione della Jupiter Ace, una vera macchina Forth-based. Non ho avuto accesso al codice sorgente della versione Ace, ma ho copiato il “look and feel” della grafica, aggiungendo un timer e 3 vite. Il livello dei dati è stato preso dal sito del compilatore Z88, dove Dark Star è incluso come progetto dimostrativo.



# **RESOURCES**



## **Informazioni**

Per contattarmi non esitate a visitare il mio sito e fare clic sulla scheda '[Contatti](#)'.

## **Argomenti per la Newsletter**

*Se volete partecipare alla stesura di questa newsletter e fornire argomenti per questa newsletter vi prego di contattarmi tramite il mio sito web.*

## **Siti**

Qui di seguito trovate le risorse in una manciata di siti che supportano i computer TI-99/4A e/o Geneve 9640. Non è certamente un elenco completo. Questa sezione sarà inclusa e aggiornata in tutte le prossime newsletter.

### **Siti Web / siti FTP**

<http://www.ti99iuc.it>

<http://www.atariage.com>

<http://shift838.wix.com/shift838>

<http://www.99er.net>

<http://www.harmlesslion.com>

<http://www.mainbyte.com>

<http://www.ninerpedia.org/>

<http://www.ti99-geek.nl/>

<http://www.turboforth.net/>

<ftp://ftp.whtech.com>

new <http://www.ti99hof.org/index.html>

new <http://www.ti99ers.org/unsung/>

new <http://ti99ers.org/modules/Inspire/remember.htm>

contiene tutti gli storici TI-99ers che sono deceduti.

### **Lista Gruppi Yahoo**

<https://groups.yahoo.com/neo/groups/TI99-4A/info>

<https://groups.yahoo.com/neo/groups/TI994A/info>

<https://groups.yahoo.com/neo/groups/Geneve9640/info>

<https://groups.yahoo.com/neo/groups/turboforth/info>

# **BBS active**

## **HeatWave BBS**

Accesso: Dial-Up e Telnet

Sistema: Geneve 9640

Software: S&T BBS Software

Località: Arizona

Contenuto: biblioteche di file TI e Geneve, messaggi di base, porte giochi ed e-mail.

Telnet: [www.heatwavebbs.com](http://www.heatwavebbs.com) port 9640 dialup: 602-955-4491 @ 8-N-1

## **The Reef Hidden**

Accesso: Dial-Up

Sistema: TI-99/4A modificato

Software: S&T BBS Software

Località: New York

Contenuto: biblioteche di file TI e Geneve, messaggi di base, porte giochi ed e-mail.

## **The Keep**

Accesso: HTTP e Telnet

Sistema: Pentium 4 con sistema operativo Windows 2000

Software: Worldgroup BBS Software (fino a 256 connessioni utente)

Località: Tigard, Oregon

Contenuto: biblioteche di file TI e Geneve, messaggi di base, porte giochi, multi-utente e giochi multiplayer ed e-mail.

Telnet: [www.thekeep.net](http://www.thekeep.net) porta 23 Web browser per <http://www.thekeep.net>

The Keep dispone di librerie di file TI, messaggi di base, e-mail, giochi porte, multi-utente e giochi multiplayer. The Keep ha anche una linea modem collegata con tutti coloro che desiderano contattare The Hidden Reef BBS da internet attraverso The Keep.

Semplicemente Telnet alla [www.thekeep.net](http://www.thekeep.net) sulla porta 23, accedi a The Keep e quindi digita **/GO DIALOUT** nel menu principale, quindi D1 per la composizione verso The Hidden Reef. E' molto semplice.

# Venditori

**SHIFT838** – Fornisce componenti TI usati come li ha acquistati. Controllate spesso cosa ho disponibile. Un sacco di articoli possono essere riutilizzati da altri utenti TI.

**Arcade Shopper** - fornisce attrezzature TI vecchie e nuove, aggiornamenti e nuove piste PCB a [www.arcadeshopper.com](http://www.arcadeshopper.com)

# Centri di riparazione

## **Richard Bell**

Riparazioni disponibili su base limitata, si prega di contattare Richard a [swim4home@verizon.net](mailto:swim4home@verizon.net) per conoscere i tempi di attesa prima di inviare qualsiasi componente da riparare

## **Tim**

Riparazioni su hardware Myarc disponibili su base limitata. Contattare Tim a [insane\\_m@hotmail.com](mailto:insane_m@hotmail.com) per i tempi di attesa o per richiedere il servizio.



