

Tiny BASIC for the 6800 & 6502 from Tom Pittman's "Company"

Editor's Note: Tom Pittman is Itty Bitty Computers. Tom is an avid computer hobbyist who is one of the more major figures in the San Francisco Bay area's Homebrew Computer Club. He is also an experienced software consultant who specializes in small-computer software and systems. We *strongly* recommend him to you because (1) we have received a number of communications praising Tom's Tiny BASIC for the 6800, (2) we have yet to receive a written or verbal complaint about the quality of Tom's products, (3) we have ample evidence that Tom consistently exhibits an unusually high level of responsiveness and good business ethics in dealing with his customers, and (4) we are delighted to see a competent systems software specialist offering excellent software products to home computer users at very nominal rates, and wish to do everything we can to encourage him and others to pursue this approach to development and distribution of good, low-cost software.

The following information is a duplication of the typewritten data sheet that Tom distributes to those wishing information about his products.

Itty Bitty Computers is dedicated to the principle of high quality, low-cost software for personal computers. Tiny BASIC is our first offering in this field; if it sells well, there will be more. We are not in the hardware business—software is all we sell. The emphasis of all our software is small size and flexibility rather than high speed. Tiny BASIC will run in as little as 3K of memory.

Tiny BASIC is a subset of Dartmouth BASIC, with only 16-bit integer arithmetic. There are only 26 variables (A-Z), no arrays, and no strings. The 12 commands (LET, PRINT, INPUT, IF...THEN, GOTO, GOSUB, RETURN, REM, RUN, LIST, CLEAR, END) are adequate for most programming needs. A machine language subroutine calling facility enables the user to extend Tiny BASIC to any degree necessary.

Single copies of Tiny BASIC cost \$5. This buys an individual license to use it, a 26-page user manual, and a hexadecimal paper tape in the manufacturer's loader format (Motorola "S" or MOS Technology ";"). A hexadecimal listing will be substituted for the tape on request for users without paper tape facilities. Programs are mailed out first class within a few days of receiving the order, but at these prices we can accept prepaid orders only. For airmail service to foreign countries please add \$2 per program. A 20% discount may be applied to quantity purchases of ten or more programs of the same version to the same address at the same time.

Tiny BASIC itself does not contain any I/O instructions; three JMPs link Tiny to the user's I/O routines. These are well documented in the manual. The primary difference between the following versions is the memory requirements (shown in parentheses), though all use memory page 00.

TB680R (0100-08FF) For MIKBUG, EXBUG, and most homebrew 6800 systems with RAM in first 3K of memory.

TB680S (0200-09FF) For Sphere; includes I/O routines.

TB680P (E000-E7FF) For AMI Prototyping Board; includes I/O routines (ACIA), and runs with on-board 1K RAM.

TB650K (0200-0AFF) For KIM and most homebrew

6502 systems with RAM in first 4K of memory.

TB650J (1000-18FF) For Jolt.

TB650W (0300-0BFF) For Apple. Includes I/O routines.

NOTE: The third digit of the version number refers to the revision level. We will ship the most recent version regardless of order specifications (e.g., we are now shipping TB682R for orders specifying TB680R).

MEMORY ALLOCATION STANDARDS FOR HOBBYISTS

by Jim Day

One of the benefits of owning your own computer is the fun of exchanging programs with other hobbyists. So far, this has been fairly easy to do. It is easy to load a BASIC program from an ASCII tape or to bootstrap a machine language tape into a "cold" machine. But this situation may not last very long. As more hobbyists begin to use ROM-resident operating systems and a wider variety of peripherals, compatibility problems may become much more difficult to resolve unless standards are established for memory allocation.

I suggest that the first 16K of memory be reserved for the operating system, including resident processors and I/O handlers. Standard port numbers and handler entry points should be established for all kinds of peripherals, including the exotic ones. Allocation standards need not work a hardship on those with smaller amounts of memory, since memory modules don't have to be contiguous. That is, a 2K machine could have 1K of memory starting at address 0 and the rest starting at 16K.

Needless to say, the standard handler "entry points" would simply be instructions branching to the real entry points somewhere within the first 16K of memory. Likewise for the entry points of language processors, text editors, etc. Attempts to use unimplemented programs should cause a branch to an error routine.

6800 CAN SHARE HOBBY-STANDARD BUS WITH 8080 WITHOUT CONFLICT

M.R.S. has developed a 6800-based board that plugs into a hobbyist-standard (Altair, IMSAI/SOL/Cromemco/Polymorphic/etc.) bus. Even while installed in the bus, it will allow any also-resident 8080 to run without interference. It gains control of the system via a single instruction, and returns control to the 8080, either via the halt switch or via software instructions. The 8080 unit handles all front-panel interface.

A fully assembled and tested unit is available for \$180 with kits available at even lower cost. The mailing address for M.R.S. is Box 1220, Hawthorne, CA 90250.

"Applications Software"-- Games in Pitman's 6800 Tiny BASIC

by Carl Kelb

These games--Stars, Acey-Deucey, Trap, and Slot--were

adapted from older versions to Tom Pitman's Tiny BASIC
for the 6800. Note: A bell (control-G) was imbedded in the
print statements for each *.

```
1 PR
1000 PRINT "WHICH GAME DO YOU WANT ?"
1001 PRINT "TYPE 1 FOR STARS"
1002 PRINT "TYPE 2 FOR ACEY-DEUCEY"
1003 PRINT "TYPE 3 FOR TRAP"
1004 PRINT "TYPE 4 FOR SLOTS"
1010 INPUT Z
1020 IF Z<1 GOTO 1000
1030 IF Z>4 GOTO 1000
1040 IF Z=1 GOTO 1100
1050 IF Z=2 GOTO 2000
1060 IF Z=3 GOTO 4000
1070 IF Z=4 GOTO 5000
1100 PR
1120 PRINT "WELCON TO MY GALAXY. I'M IN CHARGE OF THE STARS HERE."
1130 PRINT "PLAY MY GAME $STARS AND GET SOME STARS FOR YOURSELF!"
1140 PR
1150 PRINT "I WILL THINK OF A WHOLE NUMBER FROM 1 TO 100."
1160 PRINT "TRY TO GUESS MY NUMBER. AFTER YOU GUESS, I"
1170 PRINT "WILL TYPE ONE OR MORE STARS (*), THE CLOSER"
1180 PRINT "YOU ARE TO MY NUMBER, THE MORE STARS I WILL TYPE."
1190 PRINT "ONE STAR (*) MEANS YOU ARE FAR AWAY FROM MY"
1200 PRINT "NUMBER, SEVEN STARS (***** MEANS YOU ARE VERY,"
1210 PRINT "VERY, VERY CLOSE TO MY NUMBER!!!"
1220 LET X=RNDD (100)+1
1230 PR
1240 PRINT "OK , STARSEEKER, I AM THINKING OF A NUMBER, START GUESSING."
1250 LET N=1
1260 PR
1270 PRINT "WHAT IS YOUR GUESS ?"
1280 INPUT G
1290 IF G=X THEN GOTO 1550
1370 LET D=G-X
1375 IF D < 0 GOTO 1377
1376 GOTO 1380
1377 LET D = D*(-1)
1380 IF D >= 44 GOTO 1500
1390 IF D >= 32 GOTO 1490
1400 IF D >= 14 GOTO 1480
1410 IF D >= 8 GOTO 1470
1420 IF D >= 4 GOTO 1460
1430 IF D >= 2 GOTO 1450
1440 PRINT " * "
1450 PRINT " * "
1460 PRINT " * "
1470 PRINT " * "
1480 PRINT " * "
1490 PRINT " * "
1500 PRINT " * "
1510 PR
1520 LET N=N+1
1530 GOTO 1260
1550 REM
1555 PRINT " ** ** ** ** "
1560 PRINT " ** ** ** "
1580 PRINT "!!!"
1590 PRINT "THAT'S IT!!! YOU GUESSED MY COSMIC NUMBER IN 'INI' GUESSES"
1600 PR
1610 PRINT "DO YOU WANT TO PLAY AGAIN ?"
1620 PRINT "TYPE 1 TO PLAY AGAIN OR 2 TO PLAY ANOTHER GAME"
1630 INPUT I
1640 IF I=0 GOTO 1610
1650 IF I=1 GOTO 1220
1670 GOTO 1000
1680 END

2000 REM: ACEY-DEUCEY
2100 PR
2101 PRINT "ACEY-DEUCEY IS PLAYED IN THE FOLLOWING MANNER:"
2102 PRINT "THE DEALER (COMPUTER) DEALS TWO CARDS FACE UP."
2103 PRINT "YOU HAVE THE OPTION TO BET OR NOT TO BET DEPENDING"
2104 PRINT "ON WHETHER OR NOT YOU FEEL THE NEXT CARD WILL HAVE"
2105 PRINT "A VALUE BETWEEN THE FIRST TWO."
2106 PRINT "IF YOU DO NOT WANT TO BET, INPUT A 0."
2110 PR
2160 N=100
2170 Q=100
2190 PRINT "YOU NOW HAVE *IQ* DOLLARS."
```

```
2195 PR
2200 GOTO 2260
2210 LET B=Q+M
2220 GOTO 2190
2240 LET D=B-R
2250 GOTO 2190
2260 PRINT "HERE ARE YOUR NEXT TWO CARDS...."
2270 LET A=(RND(14))+2
2280 IF A<2 GOTO 2270
2290 IF A>14 GOTO 2270
2300 LET B=(RND(14))+2
2310 IF B<2 GOTO 2300
2320 IF B>14 GOTO 2300
2330 IF A>B GOTO 2270
2350 IF A<11 GOTO 2400
2360 IF A=11 GOTO 2420
2370 IF A=12 GOTO 2440
2380 IF A=13 GOTO 2460
2390 IF A=14 GOTO 2480
2400 PRINT A
2410 GOTO 2500
2420 PRINT "JACK"
2430 GOTO 2500
2440 PRINT "QUEEN"
2450 GOTO 2500
2460 PRINT "KING"
2470 GOTO 2500
2480 PRINT "ACE"
2500 IF B<11 GOTO 2550
2510 IF B=11 GOTO 2570
2520 IF B=12 GOTO 2590
2530 IF B=13 GOTO 2610
2540 IF B=14 GOTO 2630
2550 PRINT B
2560 GOTO 2650
2570 PRINT "JACK"
2580 GOTO 2650
2590 PRINT "QUEEN"
2600 GOTO 2650
2610 PRINT "KING"
2620 GOTO 2650
2630 PRINT "ACE"
2650 PR
2660 PRINT "WHAT IS YOUR BET ?"
2665 INPUT M
2670 IF M<0 GOTO 2680
2675 PRINT "CHICKEN!!"
2676 PR
2677 GOTO 2260
2680 IF M<=0 GOTO 2730
2690 PRINT "SORRY, MY FRIEND, BUT YOU BET TOO MUCH"
2700 PRINT "YOU HAVE ONLY *IQ* DOLLARS TO BET"
2710 GOTO 2650
2730 LET C=(RND(14))+2
2740 IF C<2 GOTO 2730
2750 IF C>14 GOTO 2730
2760 IF C<11 GOTO 2810
2770 IF C=11 GOTO 2830
2780 IF C=12 GOTO 2850
2790 IF C=13 GOTO 2870
2800 IF C=14 GOTO 2890
2810 PRINT C
2820 GOTO 2910
2830 PRINT "JACK"
2840 GOTO 2910
2850 PRINT "QUEEN"
2860 GOTO 2910
2870 PRINT "KING"
2880 GOTO 2910
2890 PRINT "ACE"
2910 IF C>A GOTO 2930
2920 GOTO 2970
2930 IF C>B GOTO 2970
2950 PRINT "YOU WIN!!!"
2960 GOTO 2210
2970 PRINT "SORRY, YOU LOSE."
2980 IF M<0 GOTO 2240
3000 PR
```

```

3010 PRINT 'SORRY, FRIEND, BUT YOU BLEW YOUR MAD.'
3020 PRINT
3030 PRINT 'DO YOU WANT TO TRY AGAIN ?'
3040 PRINT 'TYPE 1 TO PLAY AGAIN AND 2 TO PLAY ANOTHER GAME'
3050 INPUT I
3060 IF I=0 GOTO 3040
3070 IF I>2 GOTO 3040
3080 IF I=1 GOTO 2110
3090 IF I=2 GOTO 1000
3100 END

4000 REM: TRAP
4160 PRINT 'I WILL THINK OF A NUMBER FROM 1 TO 100.'
4170 PRINT 'TRY TO GUESS MY NUMBER. ENTER TWO NUMBERS, TRYING'
4180 PRINT 'TO TRAP MY NUMBER BY YOUR TWO TRAP NUMBERS, I'LL'
4190 PRINT 'TELL YOU IF YOU HAVE TRAPPED MY NUMBER OR IF MY'
4200 PRINT 'NUMBER IS SMALLER THAN YOUR TWO TRAP NUMBERS OR'
4210 PRINT 'IF MY NUMBER IS LARGER THAN YOUR TWO TRAP NUMBERS.'
4220 PRINT 'IF I TELL YOU THAT YOU HAVE TRAPPED MY NUMBER, I'
4230 PRINT 'MEAN THAT MY NUMBER IS BETWEEN YOUR TRAP NUMBERS'
4240 PRINT 'OR - PERHAPS MY NUMBER IS THE SAME AS ONE OF YOUR'
4250 PRINT 'TRAP NUMBERS.'
4260 PR
4270 PRINT '!!!!IMPORTANT!!! IF YOU THINK YOU KNOW MY NUMBER, THEN'
4280 PRINT 'ENTER YOUR GUESS FOR *BOTH* TRAP NUMBERS.'
4290 PR
4300 LET X=(RND(100))+1
4310 PRINT 'I'M THINKING...THINKING...AH! I HAVE A NUMBER!'
4320 LET K=1
4330 PR
4340 PRINT 'FIRST TRAP NUMBER '
4350 INPUT A
4360 PRINT 'SECOND TRAP NUMBER '
4370 INPUT B
4371 LET Y=X-A
4372 IF Y<0 THEN Y=-1
4373 IF Y>0 THEN Y=1
4374 LET Z=X-B
4375 IF Z<0 THEN Z=-1
4376 IF Z>0 THEN Z=1
4380 LET T=Y+Z
4390 IF T=-2 GOTO 4430
4391 IF T=-1 GOTO 4410
4392 IF T=0 GOTO 4400
4393 IF T=1 GOTO 4410
4394 IF T=2 GOTO 4450
4400 IF A=B GOTO 4480
4410 PRINT 'MY NUMBER IS TRAPPED BY YOUR NUMBERS.'
4420 GOTO 4460
4430 PRINT 'MY NUMBER IS SMALLER THAN YOUR TRAP NUMBERS.'
4440 GOTO 4460
4450 PRINT 'MY NUMBER IS LARGER THAN YOUR TRAP NUMBERS.'
4460 LET K=K+1
4470 GOTO 4330
4480 PRINT 'YOU GOT IT IN 'K' GUESSES ! ! ! ! ! ! !'
4485 PR
4490 PRINT 'DO YOU WANT TO PLAY AGAIN ?'
4500 PRINT 'TYPE 1 IF YOU WANT TO PLAY AGAIN OR 2 IF YOU WANT ANOTHER GAME'
4510 INPUT I
4520 IF I=0 GOTO 4500
4530 IF I>2 GOTO 4500
4540 IF I=1 GOTO 4290
4550 IF I=2 GOTO 1000
4560 END

5000 REM: SLOTS
5100 PR
5120 PRINT 'THIS IS A SIMULATION OF A SLOT MACHINE USING A COMPUTER'
5130 PRINT 'EACH TIME YOU "PULL" I WILL ASK YOU IF YOU WISH TO PLAY AGAIN.'
5140 PRINT 'JUST ANSWER WITH A "Y" FOR YES AND A "N" FOR NO.'
5150 PRINT 'PLEASE PLACE 4 QUARTERS ON MY CPU FOR EACH PLAY.'
5160 LET B=0
5170 PR
5180 LET D=(RND(8))+1
5181 LET E=(RND(7))+1
5182 LET F=(RND(7))+1
5201 IF D=1 GOTO 5211
5202 IF D=2 GOTO 5213
5203 IF D=3 GOTO 5215
5204 IF D=4 GOTO 5217
5205 IF D=5 GOTO 5219
5206 IF D=6 GOTO 5221
5207 IF D=7 GOTO 5223
5208 IF D=8 GOTO 5225
5211 PRINT '    BELL '
5212 GOTO 5231
5213 PRINT '    BAR '
5214 GOTO 5231
5215 PRINT '    CHERRY'
5216 GOTO 5231
5217 PRINT '    APPLE '
5218 GOTO 5231
5219 PRINT '    LEMON '
5220 GOTO 5231
5221 PRINT '    $ '
5222 GOTO 5231
5223 PRINT '    CHERRY'
5224 LET D=D+3
5225 GOTO 5231
5226 PRINT '    CHERRY'
5227 LET D=D+3
5228 IF E=1 GOTO 5241
5229 IF E=2 GOTO 5243
5233 IF E=3 GOTO 5245
5234 IF E=4 GOTO 5247
5235 IF E=5 GOTO 5249
5236 IF E=6 GOTO 5251
5237 IF E=7 GOTO 5253
5241 PRINT '    BELL '
5242 GOTO 5261
5243 PRINT '    BAR '
5244 GOTO 5261
5245 PRINT '    CHERRY'
5246 GOTO 5261
5247 PRINT '    APPLE '
5248 GOTO 5261
5249 PRINT '    LEMON '
5250 GOTO 5261
5251 PRINT '    $ '
5252 GOTO 5261
5253 PRINT '    CHERRY'
5254 LET E=E+3
5261 IF F=1 GOTO 5271
5262 IF F=2 GOTO 5273
5263 IF F=3 GOTO 5275
5264 IF F=4 GOTO 5277
5265 IF F=5 GOTO 5279
5266 IF F=6 GOTO 5281
5267 IF F=7 GOTO 5283
5271 PRINT '    BELL '
5272 GOTO 5410
5273 PRINT '    BAR '
5274 GOTO 5410
5275 PRINT '    CHERRY'
5276 GOTO 5410
5277 PRINT '    APPLE '
5278 GOTO 5410
5279 PRINT '    LEMON '
5280 GOTO 5410
5281 PRINT '    $ '
5282 GOTO 5410
5283 PRINT '    CHERRY'
5284 LET F=F+3
5410 IF D<E GOTO 5440
5420 IF E=F GOTO 5530
5430 IF D=E GOTO 5460
5440 IF D<F GOTO 5490
5450 GOTO 5510
5460 IF D=1 GOTO 5510
5461 IF D=3 GOTO 5510
5462 IF D=5 GOTO 5510
5463 IF D=7 GOTO 5510
5470 LET B=B+5
5475 PRINT '    KENO..YOU WIN $5..TOTAL=$'B
5480 GOTO 5550
5490 LET B=B-1
5495 PRINT '    YOU HAVE LOST $1 -- TOTAL=$'B
5500 GOTO 5550
5510 LET B=B+1
5515 PRINT '    YOU HAVE WON $1 --- TOTAL=$'B
5520 GOTO 5550
5530 IF D=2 GOTO 5540
5531 IF D=4 GOTO 5545
5535 LET B=B+20
5536 PRINT '    JACKPOT...$20...TOTAL=$'B
5537 GOTO 5550
5540 LET B=B+50
5541 PRINT '    JACKPOT...$50...TOTAL=$'B
5542 GOTO 5550
5545 LET B=B+100
5546 PRINT '    JACKPOT...$100...TOTAL=$'B
5550 PRINT '    AGAIN '
5560 INPUT I
5561 IF I=1 GOTO 5550
5562 IF I=1 GOTO 5170
5565 PR
5570 PRINT 'IT'S BEEN NICE OPERATING FOR YOU - COME BACK SOON!'
5571 PR
5575 GOTO 1000
5580 END

```

Low-Cost 6800 Systems Software & Games

by Technical Systems Consultants' staff
Box 2574, W. Lafayette IN 47906

TSC is presently involved in the creation of products which are currently in high demand among computer hobbyists and other micro computer users. Up to this time there has been little or no software available for Motorola 6800 based systems other than Monitor programs in ROM. We have developed many programs both useful and fun which allow the system builder to utilize his creation to its fullest extent. The software listings which we offer implement a variety of user and system type functions. The programs have been written in 6800 assembly language and assembled to run on Motorola and AMI 6800 based systems and utilize I/O routines contained in the MIKBUG* monitor ROM. All references to these external routines are clearly marked, however, facilitating conversion to other I/O routines. The software listings include a fully commented source listing, a hexadecimal machine code dump, sample output, and complete instructions for use. Because software "bugs" are bound to occur regardless of the degree of testing we offer a limited warranty. This 90-day warranty is limited to replacement of the original software listing or providing a patch at the discretion of TSC.

For those requiring the service, all of our routines can be assembled at a custom address or with user supplied I/O routines for an extra charge.

New products are constantly being developed by TSC. These include a Micro BASIC interpreter, a scientific floating point package, a business and accounting system, graphics games, and an 8080 emulator, among others. We also plan to offer some of our programs on "Kansas City" standard cassettes. Hardware items being developed and tested for the 6800 based system include a cassette interface system, A/D and D/A boards, a high speed arithmetic processor and other general purpose items. All of these products will be available when announced in our advertising.

We can only offer what the hobbyists want, so let us know what your needs are in both hardware and software.

*MIKBUG is a registered trademark of Motorola, Inc.

1. HANGMAN: The old word guessing game. Easily modified with your own word list. Requires 640 Bytes. \$3.25
2. ACEY-DUCEY: A card game played against the computer. Bet and try to break the bank! Requires 1K Bytes. \$3.25
3. CRAPS: A real casino craps game. Match your luck against the computer and try to win money. Requires 1K Bytes. \$3.25
4. FLOATING POINT PACKAGE: Full floating point capability. 9 digits of accuracy with exponent range, -99 to +99. Four routines for add, subtract, multiply, and divide are all included. Requires 512 Bytes. \$5.00
5. SPACE VOYAGE: Similar to the famous STAR TREK with only a few limitations. Every game is a different adventure! Requires 4K Bytes. \$10.00
6. KLINGON CAPTURE: A smaller space simulation game, but, has many of the same surprises. 2K Bytes \$4.75
7. STOCKMARKET: Similar to the popular board game of the same name. Simulates real WALL STREET action. Requires 1.5 Bytes. \$3.50
8. LINE EDITOR: Allows you to create a file in memory and then completely edit it. Commands are: NEW ADD, INSERT, DELETE, SEARCH, LIST, MOVE, PRINT. Will run in only 512 Bytes! \$4.00

9. RANDOM NUMBER GENERATOR: Here is a routine which is an absolute must for writing your own game programs. Requires 60 Bytes. \$1.50
10. MASTERMIND: Test your logical abilities; An intricate guessing game requiring both skill and logic. Requires 512 Bytes. \$3.00
11. CARD SHUFFLE AND DEAL: Two very useful routines. Includes a driver routine to print out 4 hands of 13 cards each. Requires 512 Bytes. \$2.75
12. NUMBER GUESS I: Try to guess the number the computer is thinking of! Requires 256 Bytes. \$1.50
13. NUMBER GUESS II: A more advanced number guessing game. Requires 512 Bytes. \$2.00
14. HURKLE: Try to find the hiding Hurkle relying upon clues given by the computer. Requires 640 bytes \$2.00
15. ROVER: Find and catch the Rover with the aid of hints supplied by the computer. Requires 1K Bytes. \$2.50
16. SWITCH: Correctly arrange a random string of digits in the fewest possible moves. Requires 512 Bytes. \$2.00
17. CHOMP: A 2 player game which resembles a two dimensional "NIM" game. Requires 512 Bytes. \$2.00
18. SUBROUTINE PACKAGE: A special package of very useful subroutines selected by the staff of TSC. This package could save you many hours when writing your own programs. \$3.00

**SPECIAL PACKAGE DEALS....Supplied in a 3-ring binder

- | | |
|---|---------|
| I. Contains programs 1,2,3,6,9,10 | \$13.50 |
| II. Contains programs 1,2,3,9,10,11,12,13,14,15,16,17 | \$18.95 |
| III. Contains programs in I and II plus 4 and 8 | \$29.50 |

*****SPECIAL ANNOUNCEMENT*****

At last there exists a valuable service to the computer hobbyist--THE PROGRAM OF THE MONTH CLUB-- For only \$2.00 you will get a one year membership. You will receive a monthly bulletin describing the main selection as well as many alternates. Members will receive a 15% discount on the featured program. THERE IS NO OBLIGATION TO BUY ANYTHING! If you join now, you will receive free our Random Number Generator.

[\$1.00 handling charge for orders under \$10. Add 5% for First Class Mail. Indiana Residents add 4% sales tax.]

PITTSBURGH CLUB HAS 50 MEMBERS & GROWING FAST

There is a club in Pittsburgh. It has about 50 members and is still growing; all sorts of machines and hardware; anybody in Western Pennsylvania, Eastern Ohio or West Virginia who is interested can write:

Pittsburgh Area Computer Club
400 Smithfield St.
Pittsburgh, PA 15205

Or Call:

Eric Liber (Pres.) (412) 276-6546 Nite
Fred Kitman (Treas.) (412) 391-3800 Day

Tiny BASIC for the 6800 & 6502 from Tom Pittman's "Company"

Editor's Note: Tom Pittman is Itty Bitty Computers. Tom is an avid computer hobbyist who is one of the more major figures in the San Francisco Bay area's Homebrew Computer Club. He is also an experienced software consultant who specializes in small-computer software and systems. We *strongly* recommend him to you because (1) we have received a number of communications praising Tom's Tiny BASIC for the 6800, (2) we have yet to receive a written or verbal complaint about the quality of Tom's products, (3) we have ample evidence that Tom consistently exhibits an unusually high level of responsiveness and good business ethics in dealing with his customers, and (4) we are delighted to see a competent systems software specialist offering excellent software products to home computer users at very nominal rates, and wish to do everything we can to encourage him and others to pursue this approach to development and distribution of good, low-cost software.

The following information is a duplication of the typewritten data sheet that Tom distributes to those wishing information about his products.

Itty Bitty Computers is dedicated to the principle of high quality, low-cost software for personal computers. Tiny BASIC is our first offering in this field; if it sells well, there will be more. We are not in the hardware business—software is all we sell. The emphasis of all our software is small size and flexibility rather than high speed. Tiny BASIC will run in as little as 3K of memory.

Tiny BASIC is a subset of Dartmouth BASIC, with only 16-bit integer arithmetic. There are only 26 variables (A-Z), no arrays, and no strings. The 12 commands (LET, PRINT, INPUT, IF...THEN, GOTO, GOSUB, RETURN, REM, RUN, LIST, CLEAR, END) are adequate for most programming needs. A machine language subroutine calling facility enables the user to extend Tiny BASIC to any degree necessary.

Single copies of Tiny BASIC cost \$5. This buys an individual license to use it, a 26-page user manual, and a hexadecimal paper tape in the manufacturer's loader format (Motorola "S" or MOS Technology ";"). A hexadecimal listing will be substituted for the tape on request for users without paper tape facilities. Programs are mailed out first class within a few days of receiving the order, but at these prices we can accept prepaid orders only. For airmail service to foreign countries please add \$2 per program. A 20% discount may be applied to quantity purchases of ten or more programs of the same version to the same address at the same time.

Tiny BASIC itself does not contain any I/O instructions; three JMPs link Tiny to the user's I/O routines. These are well documented in the manual. The primary difference between the following versions is the memory requirements (shown in parentheses), though all use memory page 00.

TB680R (0100-08FF) For MIKBUG, EXBUG, and most homebrew 6800 systems with RAM in first 3K of memory.

TB680S (0200-09FF) For Sphere; includes I/O routines.

TB680P (E000-E7FF) For AMI Prototyping Board; includes I/O routines (ACIA), and runs with on-board 1K RAM.

TB650K (0200-0AFF) For KIM and most homebrew

6502 systems with RAM in first 4K of memory.

TB650J (1000-18FF) For Jolt.

TB650W (0300-0BFF) For Apple. Includes I/O routines.

NOTE: The third digit of the version number refers to the revision level. We will ship the most recent version regardless of order specifications (e.g., we are now shipping TB682R for orders specifying TB680R).

MEMORY ALLOCATION STANDARDS FOR HOBBYISTS

by Jim Day

One of the benefits of owning your own computer is the fun of exchanging programs with other hobbyists. So far, this has been fairly easy to do. It is easy to load a BASIC program from an ASCII tape or to bootstrap a machine language tape into a "cold" machine. But this situation may not last very long. As more hobbyists begin to use ROM-resident operating systems and a wider variety of peripherals, compatibility problems may become much more difficult to resolve unless standards are established for memory allocation.

I suggest that the first 16K of memory be reserved for the operating system, including resident processors and I/O handlers. Standard port numbers and handler entry points should be established for all kinds of peripherals, including the exotic ones. Allocation standards need not work a hardship on those with smaller amounts of memory, since memory modules don't have to be contiguous. That is, a 2K machine could have 1K of memory starting at address 0 and the rest starting at 16K.

Needless to say, the standard handler "entry points" would simply be instructions branching to the real entry points somewhere within the first 16K of memory. Likewise for the entry points of language processors, text editors, etc. Attempts to use unimplemented programs should cause a branch to an error routine.

6800 CAN SHARE HOBBY-STANDARD BUS WITH 8080 WITHOUT CONFLICT

M.R.S. has developed a 6800-based board that plugs into a hobbyist-standard (Altair, IMSAI/SOL/Cromemco/Polymorphic/etc.) bus. Even while installed in the bus, it will allow any also-resident 8080 to run without interference. It gains control of the system via a single instruction, and returns control to the 8080, either via the halt switch or via software instructions. The 8080 unit handles all front-panel interface.

A fully assembled and tested unit is available for \$180 with kits available at even lower cost. The mailing address for M.R.S. is Box 1220, Hawthorne, CA 90250.

SAVE AND LOAD MODS TO PITTMAN'S 6800 TINY BASIC

Dear Jim,

Here are modifications I've written to Tom Pittman's 6800 Tiny BASIC. The additions are confined within the 2K size for the interpreter. The Sphere 6800 user will now have the ability to SAVE and LOAD programs he has written in Pittman's Tiny BASIC. I do not think the modifications submitted infringe on the proprietary rights of Pittman's Tiny BASIC.

Sincerely,
Henry L. Kee

42-24 Colden St.
Flushing, NY 11355

MODIFICATIONS TO PITTMAN'S TINY BASIC
SAVE PROGRAM, LOAD PROGRAM, SYSTEMS BREAK

ENTER PITTMAN'S TINY BASIC WITH I/O SPHERE INTERFACES
IN LOCATIONS D8 TO FD AS PROVIDED

ENTER THE FOLLOWING MODIFICATIONS:

```

ORIGIN C0
CE 0200      TINY BASIC START ADDRESS
DF 3C        SET PTR FOR START
CE 09FF      TINY BASIC END ADDRESS
DF 3E        SET PTR FOR END
86 42        LOAD A WITH VALUE "B"
97 34        STORE INTO PART OF IDENT
BD FB91      READ BLOCK
7E 0200      GO TO START ADDRESS FOR TINY BASIC

ORIGIN 9C1
BD FE4A      BREAKPT FOR FUTURE USE
C6 91        LOAD OPTION (DECIMAL ADDRESS 2500)
F7 09EB      MODIFY I/O FOR READ
F7 09F6      OR WRITE BY VALUE IN B
97 39        STORE 2ND CHARACTER OF ACIA
DF 33        STORE 2ND CHARACTER OF ID
86 F0        SETUP ACIA ASSIGNMENT
97 38
86 FF        SET FLAG
97 3A
86 24        STORE "$" FOR INITIAL ID
97 33
CE 00C8      BEGIN ADDRESS OF TINY BASIC PARAMETERS
DF 3C        SET BUFFER START
CE 00CF      END ADDRESS OF TINY BASIC PARAMETERS
DF 3E        SET BUFFER END
BD FB00      INITIALIZE I/O
BD FB00      READ OR WRITE
DE C8        SET USER PROGRAM START PTR
DF 3C        INTO BUFFER START
DE CC        SET USER PROGRAM END PTR
DF 3E        INTO BUFFER END
BD FB00      READ OR WRITE USER BLOCK
39          EXIT
BD FE4A      SYSTEMS HALT (DECIMAL ADDRESS 2552)
C6 2D        SAVE OPTION (DECIMAL ADDRESS 2555)
              LOAD B WITH WRITE
7E 09C6      BRANCH TO ROUTINE

```

CODING FOR V3N SYSTEM

ENTER THE FOLLOWING:

```

ORIGIN 33
24 54        "$T" ID FOR TINY BASIC
ORIGIN 38
F0 50        CASSETTE 1
FF 00
00 C0        START OF BOOTSTRAP LOADER
00 FF        END OF BOOTSTRAP LOADER

```

ORIGIN A00

```

BD FB 00      WRITE OUT BOOTSTRAP
BD FB 2D
BD FE 4A

```

EXECUTE AT LOCATION A00

```

ORIGIN 33
24 42        "$B" ID FOR TINY BASIC
ORIGIN 38
F0 50
FF 00
02 00        START OF TINY BASIC
09 FF        END OF TINY BASIC

```

EXECUTE AT LOCATION A00

YOU NOW HAVE A MODIFIED VERSION OF TINY BASIC ON CASSETTE

TO LOAD PROGRAM:

ENTER THE FOLLOWING:

```

ORIGIN 33
24 54
ORIGIN 38
F0 50
FF 00
00 C0
00 FF
ORIGIN A00
BD FB 00
BD FB 91
7E 00 C0

```

EXECUTE AT LOCATION A00

TO SAVE PROGRAM:

INTERRUPT SYSTEM
EXECUTE N=USR(2555,xx,yy)

```

N      VARIABLE THAT IS USED FOR ENTRY INTO SAVE
        COULD BE ANYTHING FROM A-Z
USR    USER CALL
2555   DECIMAL ADDRESS OF SAVE
xx     SECOND CHARACTER OF ID
        FIRST CHARACTER WILL ALWAYS BE "$"
        THE VALUE GIVEN SHOULD BE IN HEX
        e.g. 65=A, 71=C, etc
yy     SECOND CHARACTER OF ACIA
        FIRST CHARACTER WILL ALWAYS BE "F0"
        THE VALUE GIVEN SHOULD BE IN HEX
        e.g. 80=50, 96=60

```

TO LOAD PROGRAM

INTERRUPT SYSTEM
EXECUTE N=USR(2500,xx,yy)

2555 DECIMAL ADDRESS OF LOAD
all other parameters are the same as above

SYSTEMS BREAK

GOTO USR(2552)

2552 DECIMAL ADDRESS OF BREAK
THIS INSTRUCTION MAY ALSO BE PART OF CODING IN PROGRAM

TO INTERRUPT SYSTEM AT THE TINY BASIC LEVEL
HIT "REPEAT KEY"

TO IGNORE LINE JUST TYPED
HIT "ESC" KEY