

SOFTWARE FEATURES

KIM-1 BANNER

Jim Zuber
20224 Cohasset #16
Canoga Park, Ca 91306

If your KIM, SYM, or AIM system is hooked up to a printer or teletype get ready to have some fun! In the January 1979 issue of Kilobaud there was an article on page 64 called "Say it with Banner". This program prints out giant characters on your printer. There were three problems with the program:

1. It is written in 8080 code.
2. It uses octal notation.
3. It uses almost 8K of memory.

I took the general concept of printing large characters and wrote an original program that has the following features:

1. Written in 6502 code.
2. I/O independent.
3. Uses HEX notation.
4. Only uses 2K of memory.
5. Relocatable data tables.
6. Easy user modification of character sets.

Let's talk about the I/O configuration first. Location 2004 and 2005 defines the Input character routine location for your system. If your terminal echos your input change 2003 to 4C. The character output routine location for your system is defined by locations 2007 and 2008. Your output routine must do the following:

1. Provide a line feed if necessary.
2. Provide null characters if necessary.
3. Must preserve the X, Y, and accumulator registers.

Text can be stored anywhere in memory and is defined by locations 200C and 200D. The text string can be as long as you want as long as you don't run out of memory. The data tables can be stored anywhere in memory as long as the starting address of the tables is stored in 2009 and 200A. SYM users will want to store the tables right after the program. The print character is defined in location 200B. Use the HEX equivalent of the ASCII character you want. The program is set up to use an @. I will explain later on how to make up your own characters or modify some of the ones I made up. To use the program start at 2000 and GO. You will see a prompt (>>). Type in the text you want

printed out. The program treats a carriage return as a space so take note. You terminate the text input with an @. If you typed in all valid characters you will see "@ o k" printed. Get your paper ready and type a carriage return to start the printing. If you type an illegal character you will see "@ " with the illegal characters sandwiched between the @ and the ". Retype the text using only legal characters. At the end of the printout the program will prompt for more text. The legal characters are A thru Z, 0 thru 9, space, c/r, and the following characters: * . - + : ! ; ? \$, . My characters are 10 rows by 35 columns. Obviously this is too big for the AIM printer. Don't worry, you can make up your own character set to work on the AIM. To create your own character set just follow these simple rules:

1. Always store ff at the end of the tables.
2. The first BYTE should be the HEX equivalent of the ASCII character.
3. The second BYTE should be the HEX number of data Bytes.
4. Carriage returns are defined by EE.
5. Store the configuration of the character in a serial manner.
6. A "print spaces" data Byte is defined by Bit 7 being set to zero and Bits 0 thru 6 set to the number of spaces you want printed. Example: 07 would print 7 spaces.
7. To print a mark (or a character) set Bit 7 to one and Bits 0 thru 6 set to the number of marks. Example: 87 would print 7 @'s.

Maybe this will help you understand a little better. In order to print an "l" (one) that is 15 columns by 7 rows wide, just put this in the tables: 31 0A EE EE 8F EE 8F EE 8F EE EE EE. The 31 is the HEX equivalent of ASCII character one. The 0A is the number of data Bytes. Then I print 2 carriage returns, 3 rows of 15 characters and 2 more carriage returns. Hope you enjoy this program. If you want to modify any of my characters you can find their location by storing the character in 0004, then call the find character subroutine. The character's location plus 2 will be stored in 0000 and 0001.

```
0010 2000      #KIM-1 BANNER PROGRAM
0015 2000      #WRITTEN BY JIM ZUBER 12/23/78
0020 2000
0025 2000      **$0
0030 0000      PNTL **$+1
0035 0001      PNTH **$+1
0040 0002      BUF1 **$+1
0045 0003      BUF2 **$+1
0050 0004      TEMP **$+1
0055 0005      TEMPX **$+1
0060 0006      TEMPY **$+1
0065 0007
0070 0007      #KIM I/O
0075 0007      BETCH =$1E5A
0080 0007      OUTCH =$1EA0
0085 0007      CRLF =$1E2F
0090 0007
0095 0007      EOS =$40      #END OF STRING CHAR
0100 0007
0105 0007      **$2000
0110 2000 4C 0E 20  STAR  JMP OVER
0115 2003 4C 2C 21  INV   JMP INPT      #INPUT ROUTINE
0120 2006 4C 34 21  OUTV  JMP OUTC      #OUT VECTOR
0125 2009 00       TBLL  .BYTE $00      #TABLE LOW
0130 200A 30       TBLH  .BYTE $20 22   #TABLE HIGH
0135 200B 40       PRCH  .BYTE $40      #PRINT CHAR
0140 200C 00       BUFL  .BYTE $00 50   #BUFFER LOW
0145 200D 40       BUHF  .BYTE $40 21   #BUFFER HIGH
0150 200E
0155 200E DB       OVER  CLD
0160 200F A0 00     LDY $0
0165 2011 20 FA 20  JSR INTB      #INPUT TEXT
0170 2014 A9 3E     LDA #'>      #PROMPT CHAR
0175 2016 20 06 20  JSR OUTV
0180 2019 20 06 20  JSR OUTV
0185 201C A9 0D     LDA $0D      #SEND A CR
0190 201E 20 06 20  JSR OUTV
```



```

0655 20F0 20 1E 21 OUT1 JSR INCP #LOOK AT DATA
0660 20F3 B1 00 LDA (PNTL),Y
0665 20F5 AA TAX #BYTE IN X
0670 20F6 20 1E 21 JSR INCP
0675 20F9 60 RTS
0680 20FA
0685 20FA AD 0C 20 INTB LDA BUFL #INITIALIZE
0690 20FD 85 02 STA BUF1 #BUFFER SUB
0695 20FF AD 0D 20 LDA BUFL
0700 2102 85 03 STA BUF2
0705 2104 60 RTS
0710 2105
0715 2105 AD 09 20 INTP LDA TBLI #INITIALIZE
0720 2108 85 00 STA PNTL #POINTER SUB
0725 210A AD 0A 20 LDA TBLH
0730 210D 85 01 STA PNTH
0735 210F 60 RTS
0740 2110
0745 2110 18 INCB CLC #INCREMENT
0750 2111 A5 02 LDA BUF1 #BUFFER SUB
0755 2113 69 01 ADC #1
0760 2115 85 02 STA BUF1
0765 2117 A5 03 LDA BUF2
0770 2119 69 00 ADC #0
0775 211B 85 03 STA BUF2
0780 211D 60 RTS
0785 211E
0790 211E 18 INCP CLC #INCREMENT
0795 211F A5 00 LDA PNTL
0800 2121 69 01 ADC #1 #POINTER SUB
0805 2123 85 00 STA PNTL
0810 2125 A5 01 LDA PNTH
0815 2127 69 00 ADC #0
0820 2129 85 01 STA PNTH
0825 212B 60 RTS
0830 212C
0835 212C 84 06 INPT STY TEMPY #SAVE Y
0840 212E 20 5A 1E JSR GETCH #GET A CHAR
0845 2131 A4 06 LDY TEMPY
0850 2133 60 RTS
0855 2134
0860 2134 48 OUTC PHA #SAVE CHAR
0865 2135 86 05 STX TEMPX #AND X AND Y
0870 2137 84 06 STY TEMPY
0875 2139 C9 0D CMP #0D
0880 213B D0 06 BNE CONT #IS IT A C/R?
0885 213D 20 2F 1E JSR CRLF
0890 2140 4C 46 21 JMP RESTOR #GET BACK THE REGS AND RETURN
0895 2143 20 A0 1E CONT JSR OUTCH #OTHERWISE USE KIM OUTPUT.
0900 2146 A6 05 RESTOR LDX TEMPX
0905 2148 A4 06 LDY TEMPY
0910 214A 68 PLA #RETORE THE ACC.
0915 214B 60 RTS #AND RETURN
0920 214C FINISH .END

```

```

2200 20 07 EE EE EE EE EE EE EE 41 20 A3 EE A3 EE A3 2400 84 EE 84 1B 84 EE A3 EE A3 EE 50 23 A3 EE A3 EE
2210 EE 0F 85 0A 85 EE 0F 85 0A 85 EE 0F 85 0A 85 EE 2410 A3 EE 11 85 08 85 EE 11 85 08 85 EE 11 85 08 85
2220 0F 85 0A 85 EE A3 EE A3 EE A3 EE 42 2B A3 EE A3 2420 EE 11 85 08 85 EE 11 92 EE 11 92 EE 11 92 EE 51
2230 EE A3 EE 85 0A 85 0A 85 EE 85 0A 85 0A 85 EE 85 2430 20 A3 EE A3 EE 84 1B 84 EE 84 1B 84 EE 84 1B 84
2240 0A 85 0A 85 EE 85 0A 85 0A 85 EE 01 A1 EE 03 8D 2440 EE 84 02 84 15 84 EE A3 EE A3 EE 04 84 EE 02 84
2250 03 8D EE 04 8B 05 8B EE A3 22 A3 EE A3 EE A3 EE 2450 EE 52 21 A3 EE A3 EE A3 EE 11 85 08 85 EE 11 85
2260 85 19 85 EE 85 19 85 EE 85 19 85 EE 85 19 85 EE 2460 08 85 EE 11 85 08 85 EE 96 08 85 EE A3 EE 11 92
2270 88 13 88 EE 88 13 88 EE 88 13 88 EE 44 1F A3 EE 2470 EE 11 92 EE 53 36 04 81 0F 8B EE 03 82 0E 8D EE
2280 A3 EE A3 EE 85 19 85 EE 85 19 85 EE 85 19 85 EE 2480 01 84 0D 90 EE 85 08 85 09 85 EE 85 08 85 09 85
2290 85 19 85 EE 02 9F EE 03 9D EE 04 9B EE 45 2C A3 2490 EE 85 08 85 09 85 EE 85 08 85 09 85 EE 01 93 0A
22A0 EE A3 EE A3 EE 85 0A 85 0A 85 EE 85 0A 85 0A 85 24A0 84 EE 03 8F 0C 82 EE 04 8D 0D 81 EE 54 19 EE 1E
22B0 EE 85 0A 85 0A 85 EE 85 0A 85 0A 85 EE 85 0A 85 24B0 85 EE 1E 85 EE 1E 85 EE A3 EE A3 EE A3 EE 1E 85
22C0 0A 85 EE 85 19 85 EE 85 19 85 EE 46 25 A3 EE A3 24C0 EE 1E 85 EE 1E 85 EE 55 14 A3 EE A3 EE A3 EE 85
22D0 EE A3 EE 0F 85 0A 85 EE 0F 85 0A 85 EE 0F 85 0A 24D0 EE 85 EE 85 EE 85 EE A3 EE A3 EE A3 EE 56 1C 18
22E0 85 EE 0F 85 0A 85 EE 0F 85 0A 85 EE 1E 85 EE 1E 24E0 8B EE 11 8A EE 0B 89 EE 05 89 EE 88 EE 88 EE 05
22F0 85 EE 47 26 A3 EE A3 EE A3 EE 85 19 85 EE 85 19 24F0 89 EE 0B 89 EE 11 8A EE 18 8B EE 57 18 A3 EE A3

2300 85 EE 85 08 84 0D 85 EE 85 08 84 0D 85 EE 91 0A 2500 EE 8A EE 07 8D EE 11 92 EE 11 92 EE 07 8D EE 8A
2310 88 EE 91 0A 88 EE 91 0A 88 EE 48 18 A3 EE A3 EE 2510 EE A3 EE A3 EE 58 2C 88 15 86 EE 05 86 0F 86 EE
2320 A3 EE 0F 85 EE 0F 85 EE 0F 85 EE 0F 85 EE A3 EE 2520 08 86 09 86 EE 08 86 03 86 EE 0E 89 EE 0E 89 EE
2330 A3 EE A3 EE 49 1F EE 85 19 85 EE 85 19 85 EE 85 2530 08 86 03 86 EE 08 86 09 86 EE 05 86 0F 86 EE 88
2340 19 85 EE A3 EE A3 EE A3 EE 85 19 85 EE 85 19 85 2540 15 86 EE 59 1C 1A 89 EE 17 88 EE 14 86 EE 12 85
2350 EE 85 19 85 EE 4A 1A 04 87 EE 03 88 EE 02 89 EE 2550 EE 94 EE 94 EE 12 85 EE 14 86 EE 17 88 EE 1A 89
2360 84 EE 84 EE 84 EE 84 EE 02 A1 EE 03 A0 EE 04 9F 2560 EE 5A 34 86 18 85 EE 89 15 85 EE 85 01 86 12 85
2370 EE 4B 2A A3 EE A3 EE 0F 85 EE 0D 84 01 84 EE 0B 2570 EE 85 04 86 0F 85 EE 85 07 86 0C 85 EE 85 0A 86
2380 84 05 84 EE 09 84 09 84 EE 07 84 0D 84 EE 05 84 2580 09 85 EE 85 0D 86 06 85 EE 85 10 86 03 85 EE 85
2390 11 84 EE 03 84 15 84 EE 01 84 19 84 EE 4C 14 A3 2590 13 8B EE 85 16 88 EE 30 1C A3 EE A3 EE A3 EE 85
23A0 EE A3 EE A3 EE 85 EE 85 EE 85 EE 85 EE 85 EE 85 25A0 19 85 EE 85 19 85 EE 85 19 85 EE 85 19 85 EE A3
23B0 EE 85 EE 4D 18 A3 EE A3 EE 17 8C EE 0C 8E EE 8F 25B0 EE A3 EE A3 EE 31 0D EE EE EE A3 EE A3 EE A3 EE
23C0 EE 8F EE 0C 8E EE 17 8C EE A3 EE A3 EE 4E 19 A3 25C0 EE EE EE EE 32 30 94 0A 85 EE 94 0A 85 EE 94 0A
23D0 EE A3 EE 1A 89 EE 13 8A EE 0D 89 EE 0D 89 EE 06 25D0 85 EE 85 0A 85 0A 85 EE 85 0A 85 EE 85 0A
23E0 89 EE 89 EE A3 EE A3 EE 4F 20 A3 EE A3 EE 84 1B 25E0 85 0A 85 EE 85 0A 85 0A 85 EE 85 0A 85 EE 85 0A
23F0 84 EE 84 1B 84 EE 84 1B 84 EE 84 1B 84 EE 84 1B 25F0 94 EE 85 0A 94 EE 33 30 85 0A 85 0A 85 EE 85 0A

```

2600

```

3400 85 0A 85 EE 85 0A 85 0A 85 EE 85 0A 85 0A 85 EE
3410 85 0A 85 0A 85 EE 85 0A 85 0A 85 EE 85 0A 85 0A
3420 85 EE A3 EE A3 EE A3 EE 34 1B 10 93 EE 10 93 EE
3430 10 93 EE 10 85 EE 10 85 EE 10 85 EE A3 EE A3 EE
3440 A3 EE 10 85 EE 35 30 85 0A 94 EE 85 0A 94 EE 85
3450 0A 94 EE 85 0A 85 0A 85 EE 85 0A 85 0A 85 EE 85
3460 0A 85 0A 85 EE 85 0A 85 0A 85 EE 94 0A 85 EE 94
3470 0A 85 EE 94 0A 85 EE 36 2A A3 EE A3 EE A3 EE 85
3480 08 85 0C 85 EE 85 08 85 0C 85 EE 85 08 85 0C 85
3490 EE 85 08 85 0C 85 EE 92 0C 85 EE 92 0C 85 EE 92
34A0 0C 85 EE 37 1B 1B 88 EE 1B 88 EE 1B 88 EE 1E 85
34B0 EE 1E 85 EE 1E 85 EE 1E 85 EE A3 EE A3 EE A3 EE
34C0 3B 24 A3 EE A3 EE A3 EE 85 0A 85 0A 85 EE 85 0A
34D0 85 0A 85 EE 85 0A 85 0A 85 EE 85 0A 85 0A 85 EE
34E0 A3 EE A3 EE A3 EE 39 23 11 92 EE 11 92 EE 11 92
34F0 EE 11 85 08 85 EE 11 85 08 85 EE 11 85 08 85 EE

```

2700

```

3500 11 85 08 85 EE A3 EE A3 EE A3 EE 0D 07 EE EE EE
3510 EE EE EE EE 2A 34 EE 09 82 06 82 06 82 EE 0B 82
3520 04 82 04 82 EE 0D 82 02 82 02 82 EE 0F 86 EE 09
3530 92 EE 0F 86 EE 0D 82 02 82 02 82 EE 08 82 04 82
3540 04 82 EE 09 82 06 82 06 82 EE 2E 0D EE EE EE EE
3550 85 EE 85 EE 85 EE EE EE 2D 1E 0F 85 EE 0F 85
3560 EE 0F 85 EE 0F 85 EE 0F 85 EE 0F 85 EE 0F 85 EE
3570 0F 85 EE 0F 85 EE 0F 85 EE 2B 1C EE 0F 85 EE 0F

```

27

```

3580 85 EE 0F 85 EE 05 99 EE 05 99 EE 05 99 EE 0F 85
3590 EE 0F 85 EE 0F 85 EE 3A 16 EE EE EE EE 05 85 0D
35A0 85 EE 05 85 0D 85 EE 05 85 0D 85 EE EE EE EE 21
35B0 13 EE EE EE EE 85 03 9B EE 85 03 9B EE 85 03 9B
35C0 EE EE EE EE 3B 16 EE EE EE EE 05 85 0E 85 EE 04
35D0 86 0E 85 EE 02 8B 0E 85 EE EE EE EE 3F 27 17 8C
35E0 EE 17 8C EE 17 8C EE 1E 85 EE 85 02 8B 0C 85 EE
35F0 85 02 8B 0C 85 EE 85 02 8B 0C 85 EE 0D 96 EE 0D

```

3600

```

3600 96 EE 0D 96 EE 24 30 02 85 08 92 EE 02 85 08 92
3610 EE 02 85 08 92 EE 02 85 08 85 08 85 EE A3 EE A3
3620 EE 02 85 08 85 08 85 EE 02 92 08 85 EE 02 92 08
3630 85 EE 02 92 08 85 EE 2C 0F EE EE EE EE 03 85 EE
3640 02 86 EE 88 EE EE EE 28 16 EE EE EE EE A3 EE
3650 A3 EE A3 EE 85 19 85 EE 85 19 85 EE 85 19 85 EE
3660 29 16 85 19 85 EE 85 19 85 EE 85 19 85 EE A3 EE
3670 A3 EE A3 EE EE EE EE EE FF CD 00 40 5E 40 C0 30
3680 4E 1A 40 40 CF 75 C4 82 4F 6F 46 00 56 76 06 00
3690 96 75 52 40 DF C6 76 A0 6E 77 04 42 5F FF C6 80
36A0 15 3E 40 44 57 F4 07 37 F6 4F 40 04 6B EE E7 02
36B0 F7 E2 40 30 86 37 C3 05 C7 76 62 01 96 9F E6 62
36C0 47 5D 44 01 76 23 04 61 47 D3 02 01 B2 3E 00 00
36D0 97 F6 12 A0 F7 D7 40 00 97 6C 82 40 E7 F6 00 8B
36E0 B7 64 44 21 BA 0E 02 01 0F FD 00 01 A9 C1 00 00
36F0 FF FE 06 C0 76 F5 C6 D0 FF D6 00 40 A7 FC 42 60

```

3700

EDITORS NOTE: The disassembler program was originally written for the Apple and appeared in Doctor Dobbs Journal (Sept 76). It has been modified for KIM by Bob Kurtz and your editor. Bob Kurtz wrote the article.

KIM-1 "DISASSEMBLER" PROGRAM

Bob Kurtz
Micro-Z Co
P O Box 2426
Rolling Hills Ca 90274

PRELIMINARY:

The purpose of the disassembler is to take any program that has been entered into memory in the KIM-1, and to print-out an "object" code and a "source" code listing of this program - to permit analysis and modification, if desired. In a sense, it takes a completed program and reconstructs the assembly language format - or "disassembles" the program.

The following is a sample of the print-out format:

Address	Object Code	Source Code
23BC-	E8	INY
23BD-	A9 53	LDA #53
23BF-	85 01	STA 01
23C1	91 7E	STA (7E),Y
23C3-	4C 64 1C	JMP 1C64

```

0020 2000 ;DISASSEMBLER PROGRAM FOR THE 6502
0030 2000 ;WRITTEN BY STEVE WOZNAK & ALLEN BAUM
0040 2000 ;AND PUBLISHED IN DOCTOR DOBBS JOURNAL
0050 2000 ;SEPT 1976
0060 2000 ;

```

```

0070 2000 ;
0072 0000 PCL **#+1
0073 0001 PCH **#+1
0075 0002 COUNT **#+1
0080 0003 FORMAT **#+1
0090 0004 LENGTH **#+1
0100 0005 LMNEM **#+1
0110 0006 RMNEM **#+1
0142 0007 YSAVE **#+1
0150 0008 ;
0160 0008 ;KIM I/O TO FOLLOW
0170 0008 PRNBYT **#+1E3B
0180 0008 OUTCH **#+1EA0
0190 0008 CRLF **#+1E2F
0191 0008 CLEAR **#+1C64
0192 0008 OUTSP **#+1E9E
0200 0008 ;
0210 0008 ;**+$2000
0211 2000 20 0F 20 START JSR DSMBL
0212 2003 20 9E 1E JSR OUTSP
0213 2006 20 9E 1E JSR OUTSP
0214 2009 20 9E 1E JSR OUTSP
0215 200C 4C 64 1C JMP CLEAR
0220 200F A9 0D DSMBL LDA #13
0230 2011 85 02 STA COUNT

```

;COUNT FOR 13 INSTR. DSMBLY.

The Address and Object Code columns are the standard listings for the program under scrutiny. You will notice that the disassembler has arranged the code listing by one, two, or three byte commands and has printed the address column accordingly.

The Source Code columns contain the MOS Technology 650X Mnemonic abbreviations for the command - and the Operand listing. The following is an explanation of the address mode for the various operands:

Operand	Address Mode
blank	Accumulator, Implied
#53	Immediate
01	Zero Page
01,X 01,Y	Zero Page, indexed by X or Y
(7E),Y	Indirect Indexed
(7E,X)	Indexed Indirect
1C64	Absolute of Branch
(1C64)	Indirect
1C64,X	Absolute (indexed by X)
1C64,Y	Absolute (indexed by Y)

PROCEDURE:

1. Load the starting address of the program to be disassembled into locations 0000 (Low byte) and 0001 (High byte).
2. Go to location 2000
3. Press "G" on terminal

The "disassembler" will now print-out the first 13 commands of the program under scrutiny. At the end of this print-out, simply press "G" again and the next 13 commands will be printed out. Continuing to press "G" whenever the program stops, will step you through the entire program under investigation.

The program stops after each 13 commands. If you wish to modify this, change the byte in location 2010 from \$0D (13 decimal) to any number up to \$FF (256 decimal).

If portions of the disassembled program do not appear to make sense, these may be "look-up" tables within the program. As an example, the disassembler can be used to "disassemble" the disassembler program! Addresses \$2000 to \$211A will print out properly since these contain the body of the program commands. However, locations \$211B to \$21F9 contain the tables for all the mnemonics and symbols and will print-out gibberish.