Machine Language Menu Master

Add the professional *touch of AppleWorks-like menus to your machine language programs, with this easy-to-use routine.*

Menu Master is a machine language subroutine that allows you to create AppleWorks-like menus easily and add them to your machine language programs. It can handle up to 23 menu entries, and more if you use submenus.

The type of menu created by Menu Master is usually called an "inverse bar" menu, because the selected menu item is displayed in inverse. The user can move the inverse bar up and down through the menu by pressing the arrow keys, or by typing in the number of a menu item. The user accepts a menu selection by pressing Return. Menu Master takes care of all the details involved in handling the menu, including numbering the items. All you must do is set up a few parameters and call the Menu Master subroutine.

USING MENU MASTER

To use Menu Master in your own programs, you must set up a short parameter list. **Table 1** shows the format of the parameter list. The first three parameters indicate the default menu item, the horizontal position, and the vertical position of the menu, respectively. The fourth parameter indicates the number of items in the menu. The fourth parameter is followed by a list of addresses, one for each menu item. Each address must point to the text for a menu item. The text must be stored in ASCII-terminated format, which means that the last character of the string should have the high bit cleared. The Merlin assembler uses the DCI psuedo-opcode to create this type of text string, the S-C Macro Assembler uses .AT, and other assemblers have their own syntaxes (check your manual). If you can't create ASCII-terminated strings with your assembler, you may terminate the strings with a zero. The routine will still work, but it uses an extra byte of memory. Another alternative is to hand-code the last byte of the string using the HEX psuedo-op. The text may be upper-case or lower-case; Menu Master automatically converts it to upper-case when it is highlighted, to avoid problems with MouseText characters.

Menu Master takes care of all the details involved in handling the menu, including numbering the items.

Once you have set up the parameter list according to your needs, your program must load the Accumulator with the low byte of the address of the parameter list, and the X-Register with the high byte of the address of the parameter list. Call the Menu Master subroutine with a JSR. When it returns, the Accumulator will contain the number of the menu item selected. If Escape has been pressed, the Accumulator will contain $FF. If there is an error in the parameter list (e.g., too many entries), the Accumulator will contain a $00.

ENTERING THE PROGRAM

The listings for Menu Master appear in the Program Listings section at the end of the magazine.

To enter the Menu Master demonstration (**Listing 1**), either type the assembly source code into your assembler and assemble the program, or enter the hexadecimal code from the Monitor and save it with:

BSAVE MENU,A$8000,L$281

TABLE 1: Parameter List Format

	Parameters	Function
LABEL	HEX S	S is the starting menu item, the one that will be highlighted when the menu is first displayed.
	HEX X	X is the starting horizontal position of the menu (HTAB).
	HEX Y	Y is the starting vertical position of the menu (VTAB).
	HEX *N*	*N* is the number of menu items.
	DA AD1	AD1 is the address of the first item's text.
	...	Addresses for each item go here
	DA AD*n*	AD*n* is the address of the last item's text.

For help with entering *Nibble* programs, see "A Welcome to New *Nibble* Readers" at the beginning of this issue.

To run the demo, simply type **BRUN MENU**. A short three-item menu is displayed, and you can choose from Quit to BASIC, A Brief Synopsis, and Something Else. Quit to BASIC returns you to Applesoft. A Brief Synopsis displays a short message about Menu Master, and Something Else displays the phrase "Something Else..."

To extract the Menu Master subroutine itself from **Listing 1**, you should delete **lines 1-75**. The resulting assembly source code can then be added to your own program, where you can set up your own parameter lists (you must have an assembler to do this, of course).

HOW IT WORKS

The Demonstration Portion

Lines 1-72 of **Listing 1** demonstrate the use of the Menu Master subroutine in **lines 76-315**. **Lines 11-15** set up some standard equates. **Line 17** clears the screen, and then **lines 18-25** display the copyright notice and the user prompt. **Lines 26-27** position the cursor on line 4 prior to calling the Menu Master subroutine. The address of the parameter list (**lines 53-61**) is loaded into the A and X Registers, and the menu subroutine is called in **line 30**. **Lines 31-32** check whether Escape has been pressed, and if so, restart the program. **Lines 33-34** check whether the first menu item (Quit to BASIC) has been selected, and if so, the program ends by calling WRMSTRT at $3D0.

Lines 36-43 handle the second menu item by displaying a short message and then waiting for a keypress before returning to the start of the program. **Lines 44-49** are similar, but display the message corresponding to the third menu item.

Lines 53-61 are the parameter list, and **lines 62-72** contain the various messages the program uses.

The Menu Subroutine

The menu subroutine starts in **line 151** by storing the address of the parameter list in the zero page locations, PARM and PARM+1 ($FE,$FF). This address is used throughout the routine to access the data in the parameter list. **Lines 153-173** set up the routine's variables. If there are more than 23 menu items, the routine returns a zero to indicate an error condition. **Lines 158-212** display the menu on the screen, according to the data in the parameter list. When a menu item is highlighted, $3F is stored in INVFLG, and $DF is stored in MASK. These two values result in upper-case inverse letters.

Lines 213-263 handle all the possible keypresses. If an invalid key is pressed, the BELL routine (**lines 272-282**) produces the standard ProDOS beep.

Lines 289-315 contain a generic print routine that is used to display all the text for menu items and prompts. You can use it from your own program as well. To call it, load the low byte of the address of the text you want to display into the Accumulator, and the high byte of the address into the X-Register. Call the print routine with a JSR. The print routine takes into account the values of INVFLG and MASK, so you can convert lower-case to upper-case by storing a $DF in MASK, and you can display inverse letters by storing a $3F in INVFLG. To return to normal text, store a $FF in both INVFLG and MASK.

Listing 1 for Machine Language Menu Master

MENU

```
                 1    ************************
                 2    * MENU                 *
                 3    * BY CHRIS MEYER       *
                 4    * COPYRIGHT (C) 1987   *
                 5    * BY MICROSPARC, INC.  *
                 6    * CONCORD, MA  01742   *
                 7    ************************
                 8    * MERLIN ASSEMBLER
                 9    ************************
                10    * Absolute Equates
                11    WRMSTRT  =      $3D0       ;Enter BASIC
                12    HOME     =      $FC58      ;Clear the screen
                13    RDKEY    =      $FD0C      ;Read a character
                14    PRBYTE   =      $FDDA      ;Print a byte in A-reg
                15    TABV     =      $FB5B
                16    ************************
8000: 20 58 FC  17    START    JSR    HOME       ;Clear the screen
8003: A9 FB     18             LDA    #<COPYR
8005: A2 80     19             LDX    #>COPYR
8007: 20 4F 82  20             JSR    PRINT
800A: A9 17     21             LDA    #23
800C: 20 5B FB  22             JSR    TABV
800F: A9 89     23             LDA    #<TEXT
8011: A2 80     24             LDX    #>TEXT
8013: 20 4F 82  25             JSR    PRINT
8016: A9 04     26             LDA    #4
8018: 20 5B FB  27             JSR    TABV
801B: A9 51     28             LDA    #<PARMS
801D: A2 80     29             LDX    #>PARMS
801F: 20 3F 81  30             JSR    MENU       ;Call the menu subroutine
8022: C9 FF     31             CMP    #$FF
8024: F0 DA     32             BEQ    START
8026: C9 01     33             CMP    #$01
8028: D0 03     34             BNE    START1
802A: 4C D0 03  35             JMP    WRMSTRT
802D: C9 02     36    START1   CMP    #$02
802F: D0 10     37             BNE    START2
8031: 20 58 FC  38             JSR    HOME
8034: A9 AC     39             LDA    #<TEXT2
8036: A2 80     40             LDX    #>TEXT2
8038: 20 4F 82  41             JSR    PRINT
803B: 20 0C FD  42             JSR    RDKEY
803E: 4C 00 80  43             JMP    START
8041: 20 58 FC  44    START2   JSR    HOME
8044: A9 EA     45             LDA    #<TEXT3
8046: A2 80     46             LDX    #>TEXT3
8048: 20 4F 82  47             JSR    PRINT
804B: 20 0C FD  48             JSR    RDKEY
804E: 4C 00 80  49             JMP    START
                50    *
                51    * The parameter list
                52    *
8051: 01        53    PARMS    HEX    01
8052: 0A 05     54             HEX    0A05
8054: 03        55             HEX    03
8055: 5B 80     56             DA     ENTRY1
8057: 68 80     57             DA     ENTRY2
8059: 7B 80     58             DA     ENTRY3
805B: D1 F5 E9  59    ENTRY1   DCI    "Quit to BASIC"
805E: F4 A0 F4 EF A0 C2 C1 D3
8066: C9 43
8068: D6 E5 F2  60    ENTRY2   DCI    "Very Brief Synopsis"
806B: F9 A0 C2 F2 E9 E5 E6 A0
8073: D3 F9 EE EF F0 F3 E9 73
807B: D3 EF ED  61    ENTRY3   DCI    "Something Else"
807E: E5 F4 E8 E9 EE E7 A0 C5
8086: EC F3 65
8089: C1 F2 F2  62    TEXT     DCI    "Arrows to move or Return to select."
808C: EF F7 F3 A0 F4 EF A0 ED
8094: EF F6 E5 A0 EF F2 A0 D2
809C: E5 F4 F5 F2 EE A0 F4 EF
80A4: A0 F3 E5 EC E5 E3 F4 2E
80AC: CD E5 EE  63    TEXT2    ASC    "Menu Master: A menu subroutine callable"
80AF: F5 A0 CD E1 F3 F4 E5 F2
80B7: BA A0 C1 A0 ED E5 EE F5
80BF: A0 F3 F5 E2 F2 EF F5 F4
80C7: E9 EE E5 A0 E3 E1 EC EC
80CF: E1 E2 EC E5
80D3: 8D        64             HEX    8D
80D4: E6 F2 EF  65             DCI    "from machine language."
```

```
807:  ED A0 ED E1 E3 E8 E9 EE
80F:  E5 A0 EC E1 EE E7 F5 E1
0E7:  E7 E5 2E
0EA:  D3 EF ED   66   TEXT3     DCI    "Something else..."
0ED:  E5 F4 E8 E9 EE E7 A0 E5
0F5:  EC F3 E5 AE AE 2E
0FB:  CD E5 EE   67   COPYR     ASC    "Menu Master Demo"
0FE:  F5 A0 CD E1 F3 F4 E5 F2
106:  A0 C4 E5 ED EF
10B:  8D         68             HEX    8D
10C:  E2 F9 A0   69             ASC    "by Chris Meyer"
10F:  C3 E8 F2 E9 F3 A0 CD E5
117:  F9 E5 F2
11A:  8D         70             HEX    8D
11B:  C3 EF F0   71             ASC    "Copyright 1987 by MicroSPARC, Inc."
11E:  F9 F2 E9 E7 E8 F4 A0 B1
8126: B9 B8 B7 A0 E2 F9 A0 CD
812E: E9 E3 F2 EF D3 D0 C1 D2
8136: C3 AC A0 C9 EE E3 AE
813D: 8D 00      72             HEX    8D00
                 73   ********************************
                 74   * NOTE: Delete lines 1-75 to get*
                 75   * the MENU subroutine alone     *
                 76   ********************************
                 77   * Machine Language Menu *
                 78   *        Master         *
                 79   *************************
                 80   *
                 81   * Zero page equates
                 82   *
                 83   CH        =      $24
                 84   CV        =      $25
                 85   INVFLG    =      $32
                 86   PRNT      =      $F5
                 87   LEN       =      $F7
                 88   LIN       =      $F8
                 89   ENTRIES   =      $F9
                 90   CURRENT   =      $FA
                 91   CH2       =      $FB
                 92   MASK      =      $FC
                 93   POS       =      $FD
                 94   PARM      =      $FE
                 95   *
                 96   * Absolute equates
                 97   *
                 98   KBD       =      $C000      ;Keyboard data
                 99   STROBE    =      $C010      ;Keyboard strobe
                 100  SPKR      =      $C030      ;Click speaker
                 101  VTABZ     =      $FC24      ;Vtab to cv
                 102  WAIT      =      $FCA8      ;Wait routine
                 103  COUT      =      $FDED      ;Output characte
                 104  *
                 105  * MENU: This is an automatic
                 106  * menu routine that can be called
                 107  * by machine language routines.
                 108  * Enter with A-reg,X-reg set to
                 109  * the address of the parameter
                 110  * list. The parameter list is
                 111  * constructed like this:
                 112  *
                 113  *  HEX 01 (This is the current
                 114  *          menu selection)
                 115  *  HEX 0F05 (This is the X and Y
                 116  *            screen position)
                 117  *  HEX 05 (This is the number of
                 118  *          menu entries)
                 119  *  DA ENTRY1 (Address to text of
                 120  *             first menu entry)
                 121  *  DA ENTRY2
                 122  *  ...
                 123  *  DA ENTRY5
                 124  *
                 125  *  The text menu entries must
                 126  *  have the last character
                 127  *  inverted using the "DCI"
                 128  *  pseudo-opcode like this:
                 129  *
                 130  * ENTRY1 DCI "Quit"
                 131  * ENTRY2 DCI "Edit"
                 132  * ENTRY3 DCI "Save"
                 133  * ENTRY4 DCI "Load"
                 134  * ENTRY5 DCI "Print"
                 135  *
                 136  * Note: If you are using
                 137  * lowercase in your menu entries
                 138  * don't worry about the inverse
                 139  * as the routine will convert
                 140  * inverse lowercase to uppercase.
                 141  *
                 142  * Note: Be sure and do a HOME by
                 143  * JSRing to $FC58 before calling
                 144  * this routine.
                 145  *
                 146  * This routine returns with the
                 147  * A-reg containing the entry
                 148  * number selected or if "ESC" was
                 149  * pressed, a $FF is returned.
                 150  *
```

```
813F: 85 FE      151  MENU      STA    PARM
8141: 86 FF      152            STX    PARM+1
8143: A0 00      153            LDY    #$0
8145: B1 FE      154            LDA    (PARM),Y
8147: 85 FD      155            STA    POS
8149: 88         156            DEY
814A: 84 FC      157            STY    MASK
814C: A0 01      158  MENU1     LDY    #$1
814E: 84 FA      159            STY    CURRENT
8150: B1 FE      160            LDA    (PARM),Y
8152: 85 24      161            STA    CH
8154: 85 FB      162            STA    CH2
8156: C8         163            INY
8157: B1 FE      164            LDA    (PARM),Y
8159: 85 25      165            STA    CV
815B: 20 24 FC   166            JSR    VTABZ
815E: C8         167            INY
815F: B1 FE      168            LDA    (PARM),Y
8161: 85 F9      169            STA    ENTRIES
8163: C9 18      170            CMP    #24
8165: 90 03      171            BCC    MENU6
8167: A9 00      172            LDA    #$0
8169: 60         173            RTS
816A: E6 F9      174  MENU6     INC    ENTRIES
816C: A9 00      175            LDA    #$0
816E: 85 F8      176            STA    LIN
8170: C8         177            INY
8171: A5 FA      178  MENU2     LDA    CURRENT
8173: C5 FD      179            CMP    POS
8175: D0 08      180            BNE    MENU3
8177: A9 3F      181            LDA    #$3F
8179: 85 32      182            STA    INVFLG
817B: A9 DF      183            LDA    #$DF
817D: 85 FC      184            STA    MASK
817F: E6 F8      185  MENU3     INC    LIN
8181: A6 F8      186            LDX    LIN
8183: CA         187            DEX
8184: BD 1F 82   188            LDA    NUMLET,X
8187: 20 ED FD   189            JSR    COUT
818A: A9 AE      190            LDA    #"."
818C: 20 ED FD   191            JSR    COUT
818F: A9 A0      192            LDA    #$A0
8191: 20 ED FD   193            JSR    COUT
8194: B1 FE      194            LDA    (PARM),Y
8196: 48         195            PHA
8197: C8         196            INY
8198: B1 FE      197            LDA    (PARM),Y
819A: C8         198            INY
819B: AA         199            TAX
819C: 68         200            PLA
819D: 20 53 82   201            JSR    PRINTA
81A0: A9 8D      202            LDA    #$8D
81A2: 20 ED FD   203            JSR    COUT
81A5: A5 FB      204            LDA    CH2
81A7: 85 24      205            STA    CH
81A9: A9 FF      206            LDA    #$FF
81AB: 85 32      207            STA    INVFLG
81AD: 85 FC      208            STA    MASK
81AF: E6 FA      209            INC    CURRENT
81B1: A5 FA      210            LDA    CURRENT
81B3: C5 F9      211            CMP    ENTRIES
81B5: 90 BA      212            BCC    MENU2
81B7: AD 00 C0   213  MENU4     LDA    KBD
81BA: 10 FB      214            BPL    MENU4
81BC: 2C 10 C0   215            BIT    STROBE
81BF: C9 9B      216            CMP    #$9B
81C1: F0 3E      217            BEQ    MENUESC
81C3: C9 88      218            CMP    #$88
81C5: F0 3D      219            BEQ    MENUBCK
81C7: C9 95      220            CMP    #$95
81C9: F0 45      221            BEQ    MENUFWD
81CB: C9 8B      222            CMP    #$8B
81CD: F0 35      223            BEQ    MENUBCK
81CF: C9 8A      224            CMP    #$8A
81D1: F0 3D      225            BEQ    MENUFWD
81D3: C9 8D      226            CMP    #$8D
81D5: F0 23      227            BEQ    MENUACP
81D7: C9 B1      228            CMP    #$B1
81D9: 90 19      229            BCC    MENU5
81DB: C9 BA      230            CMP    #$BA
81DD: 90 0A      231            BCC    MENU7
81DF: C9 C0      232            CMP    #$C0
81E1: 90 11      233            BCC    MENU5
81E3: C9 CF      234            CMP    #$CF
81E5: B0 0D      235            BCS    MENU5
81E7: E9 06      236            SBC    #$6
81E9: 38         237  MENU7     SEC
81EA: E9 B0      238            SBC    #$B0
81EC: C5 F9      239            CMP    ENTRIES
81EE: B0 04      240            BCS    MENU5
81F0: 85 FD      241            STA    POS
81F2: D0 28      242            BNE    MENUJMP
81F4: 20 36 82   243  MENU5     JSR    BELL
81F7: 4C B7 81   244            JMP    MENU4
```

Listing 1 for Machine Language Menu Master

MENU *(continued)*

```
81FA: A0 00          MENUACP  LDY    #$0
81FC: A5 FD                   LDA    POS
81FE: 91 FE                   STA    (PARM),Y
8200: 60                      RTS
8201: A9 FF          MENUESC  LDA    #$FF
8203: 60                      RTS
8204: C6 FD          MENUBCK  DEC    POS
8206: D0 14                   BNE    MENUJMP
8208: A5 F9                   LDA    ENTRIES
820A: 85 FD                   STA    POS
820C: C6 FD                   DEC    POS
820E: D0 0C                   BNE    MENUJMP
8210: E6 FD          MENUFWD  INC    POS
8212: A5 FD                   LDA    POS
8214: C5 F9                   CMP    ENTRIES
8216: 90 04                   BCC    MENUJMP
8218: A9 01                   LDA    #$1
821A: 85 FD                   STA    POS
821C: 4C 4C 81       MENUJMP  JMP    MENU1
821F: B1 B2 B3       NUMLET   ASC    "123456789A"
8222: B4 B5 B6 B7 B8 B9 C1
8229: C2 C3 C4                ASC    "BCDEFGHIJK"
822C: C5 C6 C7 C8 C9 CA CB
8233: CC CD CE                ASC    "LMN"
                     *
                     * BELL: This routine is directly
                     * from the Apple Workbench Manual
                     * It produces a "soft" bell.
                     *
8236: A9 20          BELL     LDA    #$20
8238: 85 F7                   STA    LEN
823A: A9 02          BELL1    LDA    #$2
823C: 20 A8 FC                JSR    WAIT
823F: 8D 30 C0                STA    SPKR
8242: A9 24                   LDA    #$24
8244: 20 A8 FC                JSR    WAIT
8247: 8D 30 C0                STA    SPKR
824A: C6 F7                   DEC    LEN
824C: D0 EC                   BNE    BELL1
824E: 60                      RTS
                     *
                     * PRINT: Prints characters
                     * starting at A-reg,X-reg until
                     * a positive charcter (less than
                     * $80) is encountered.
                     *
824F: A0 FF          PRINT    LDY    #$FF
8251: 84 FC                   STY    MASK
8253: 85 F5          PRINTA   STA    PRNT
8255: 86 F6                   STX    PRNT+1
8257: 98                      TYA
8258: 48                      PHA
8259: A0 00                   LDY    #$0
825B: B1 F5          P1       LDA    (PRNT),Y
825D: 10 10                   BPL    P2
825F: C9 E1                   CMP    #$E1
8261: 90 06                   BCC    P3
8263: C9 FB                   CMP    #$FB
8265: B0 02                   BCS    P3
8267: 25 FC                   AND    MASK
8269: 20 ED FD       P3       JSR    COUT
826C: C8                      INY
826D: D0 EC                   BNE    P1
826F: 09 80          P2       ORA    #$80
8271: C9 E1                   CMP    #$E1
8273: 90 06                   BCC    P4
8275: C9 FB                   CMP    #$FB
8277: B0 02                   BCS    P4
8279: 25 FC                   AND    MASK
827B: 20 ED FD       P4       JSR    COUT
827E: 68                      PLA
827F: A8                      TAY
8280: 60                      RTS

--End assembly, 641 bytes, Errors: 0
```

END OF LISTING 1

```
                 KEY PERFECT 5.0
                     RUN ON
                      MENU
   ==========================================
     CODE-5.0     ADDR# - ADDR#     CODE-4.0
     --------     -------------     --------
     1A821AC6      8000 - 804F        2A29
     A247964E      8050 - 809F        264E
     3F8ED55F      80A0 - 80EF        28E7
     46AD7787      80F0 - 813F        29FD
     78E1A067      8140 - 818F        24FA
     39D23824      8190 - 81DF        29A5
     F91A38B3      81E0 - 822F        241F
     CA5E2BD5      8230 - 827F        2625
     2A714F60      8280 - 8280          C0
     9CFBEF38 = PROGRAM TOTAL =       0281
```