ATARI CX85 NUMERICAL KEYPAD

Technical Reference Notes

This booklet contains information for advanced programmers who want to modify the keypad handler program or create their own program to read data from the keypad. The *ATARI® CX85™ Numerical Keypad Owner's Guide* contains additional information with which you should be familiar.

Modifying the Keypad Handler Program

The diskette that came with this package contains a keypad handler program which is written with the ATARI Macro Assembler AMAC. You can modify or rewrite this program with the Program-Text Editor. When your modifications are complete, reassemble the code with AMAC using a unique filename.

Hardware Notes:
Keypad Interface and Timing

The keypad uses eight signals on the controller port. Positive 5 volts (+5 v) is on pin 7, and signal ground on pin 8. A 5-bit binary code is presented on pins 1 through 5, corresponding to the signals FWD, BACK, LEFT, RIGHT and BPOT. A data valid signal, presented on pin 6, corresponds to TRIGGER. TRIGGER goes low to indicate a valid code.

Timing is as follows:

- With no key pressed, the code for the previously pressed key remains on pins 1 through 5 and TRIGGER remains high (logic 1 or True).
- When a key is pressed, the TRIGGER signal goes low (logic 0 or False) and the keycode for that key is established on pins 1 through 5.
- TRIGGER stays low as long as the key remains pressed. When the key is released, TRIGGER returns high but the keycode does not change.
- Two-key rollover handles simultaneous or multiple keystrokes. If one or more additional keys are pressed while the first key is still pressed, nothing happens; the additional keys are locked out. When the first key is released, TRIGGER goes high and the scanning electronics searches for the next active key in the sequence. TRIGGER then goes low and the new keycode is presented.

After the system receives the TRIGGER signal there is a slight delay before BPOT data is valid. This can be compensated for by inserting a delay in your program. Details about this timing difference are provided on page 5.

Software Notes

Signals generated by the keypad electronics are delivered to specific registers in your computer's memory through the joystick port. The keypad makes use of registers normally used for both joystick and paddle controllers. The sequence in which your program reads these signals and the operations your program performs on the signals are both important factors. TRIGGER must be read first to see if a key has been pressed. If a key has been pressed, it must be decoded. Four bits of the 5-bit keycode are sent to the joystick registers and the fifth bit is read through the paddle (BPOT) register. These bits must be manipulated and logically combined for your program to know which key has been pressed. Once the keypad key has been decoded, it can be associated with any key on the keyboard, or any operation for which your keypad handler is designed.

Controller Port Selection

The handler program, as written, allows the keypad to work only through joystick port 2. However, it can be rewritten to allow the keypad to work through any one of the four controller ports. You'll have to modify the handler to use alternate registers or write those registers into a new program to recognize the port.

Determining a Keypress

To determine if a key has been pressed, your program must read the OS shadow for the joystick controller port to which the keypad is connected.

For port 1 read STRIG0 ($284).

For port 2 read STRIG1 ($285).

For port 3 read STRIG2 ($286).

For port 4 read STRIG3 ($287).

Only the least significant bit is used (bit 0). If a key has been pressed, the LSB will contain a zero (0). If the bit contains a one (1) no key has been pressed. The remaining seven bits will contain zeroes.

Decoding Keys

Once the STRIG value has been read and a data-valid signal exists, the incoming 5-bit code must be decoded. To do this, you must read the hardware registers, not the OS shadows. Read the joystick byte first.

Registers PORTA ($D300) and PORTB ($D301) pick up the first four incoming bits (see Figures 1 and 2). Depending on which port is selected, the bits will come into positions 1 through 4 (bits 0-3), or positions 5 through 8 (bits 4-7). They must be shifted to bits 0 through 3, and bits 4 through 7 must be forced to zero.

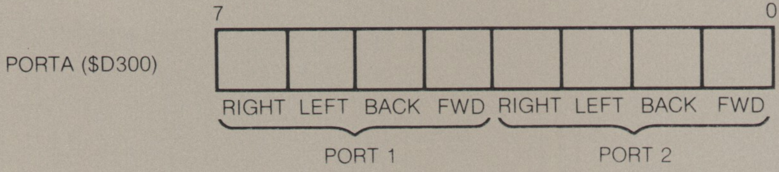


Figure 1

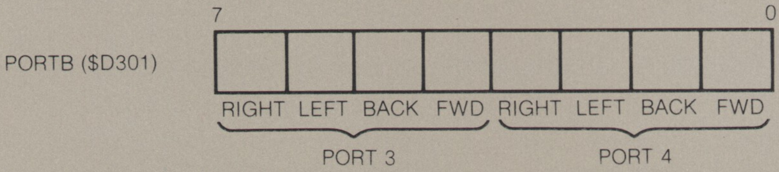


Figure 2

Note: All shifting and masking operations must be done in the accumulator—do not use memory address shift instructions.

BPOT is extracted from ALLPOT ($D208)—see Figure 3. All bits in the byte, except the bit for the desired port, must be forced to zero. The chosen BPOT bit must be complemented and shifted so that it's in the fifth position (bit 4).

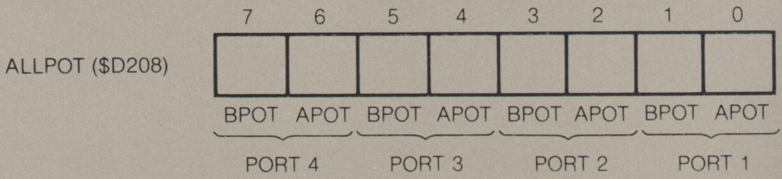


Figure 3

Finally, a logical OR must be used to integrate the BPOT value and the values of the joystick operation. The resultant binary value, represented in Figure 4, can be decoded from the truth table on pages 6-7.

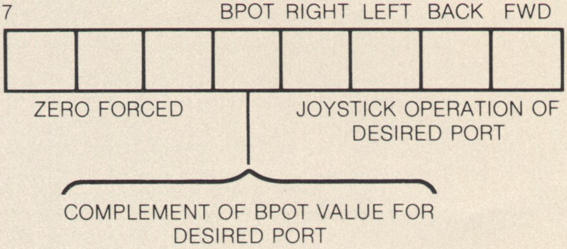

7
BPOT
RIGHT
LEFT
BACK
FWD
ZERO FORCED
JOYSTICK OPERATION OF DESIRED PORT
COMPLEMENT OF BPOT VALUE FOR DESIRED PORT

Figure 4

Timing Differences Between TRIGGER and BPOT

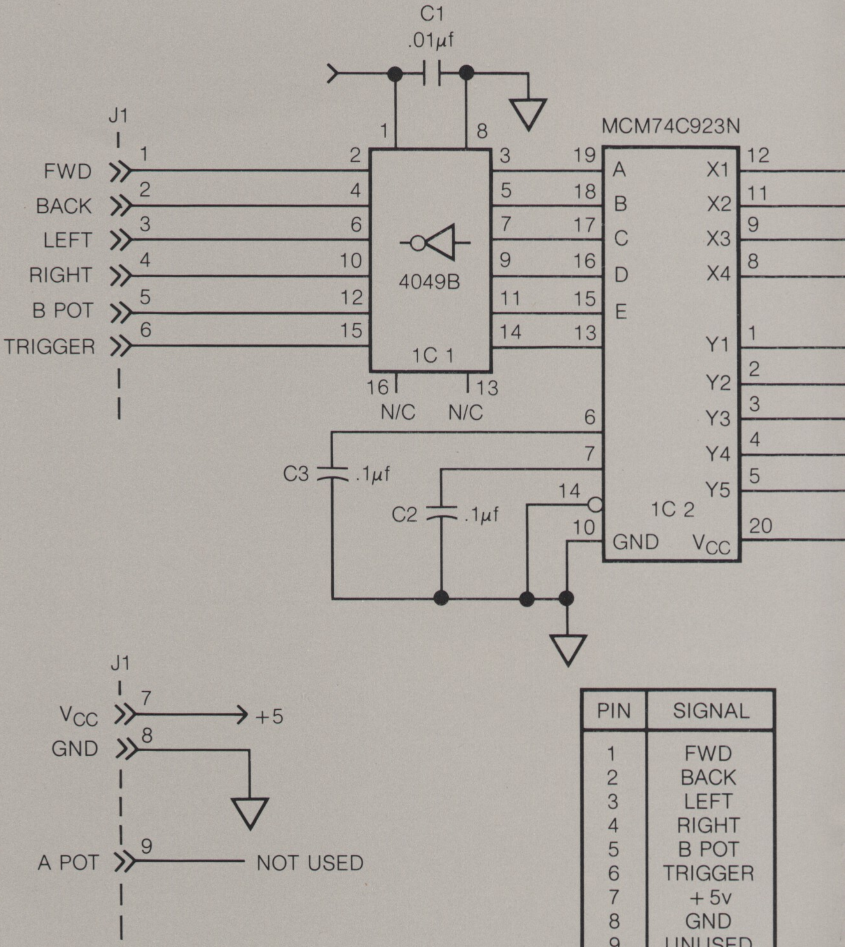
The BPOT value is input to POKEY through a resistor-capacitor delay circuit in your computer. The TRIGGER signal indicating a new keypress occurs instantly, while the BPOT signal might be delayed up to 150 microseconds. Insert an intentional delay in your program to compensate for this time difference—150 microseconds corresponds to approximately 150 NOPs—or insert a loop decrementing a register from 30 to zero.

Appending to DOS II

To use the serial ports on your ATARI 850™ Interface Module and load the keypad handler at the same time, you must append KEYPAD.OBJ to the end of the DOS II AUTORUN.SYS file. The interface module is booted automatically with the AUTORUN.SYS file. If you append the keypad handler to the end of this file, the interface module will be booted and your keypad enabled as well. This should be done to a copy of the DOS II Master Diskette using the COPY FILE command in DOS II. Please refer to the *ATARI Disk Operating System II Reference Manual* for the correct procedures.

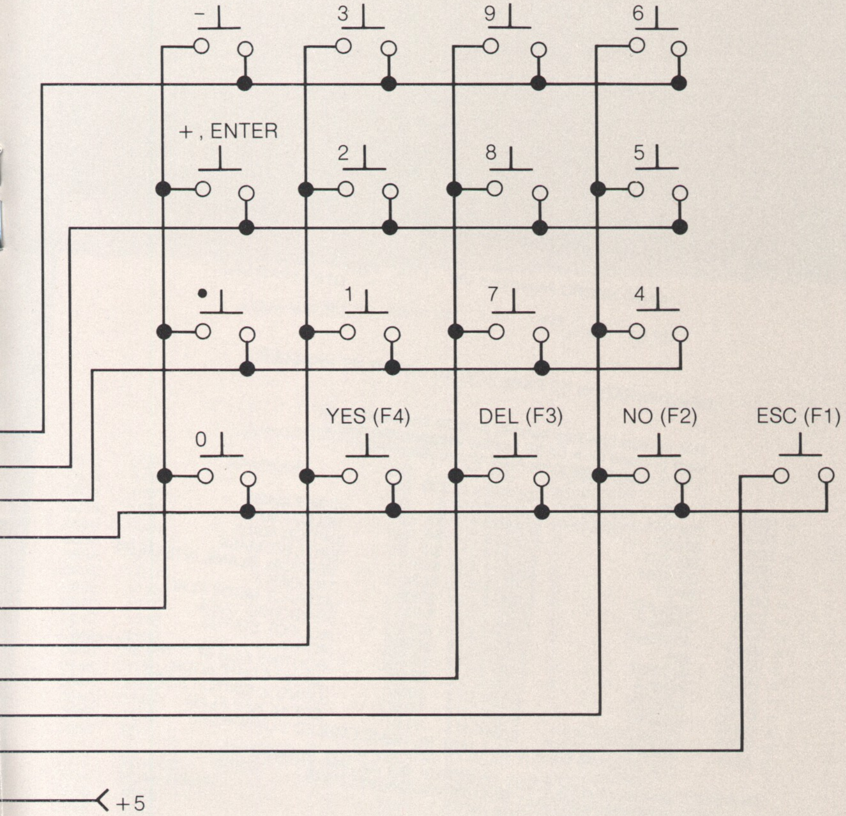
If you plan to load a keypad handler and use DOS more than once while programming, your keypad handler diskette must have a MEM.SAV file.

Schematic and Truth Table



PIN	SIGNAL
1	FWD
2	BACK
3	LEFT
4	RIGHT
5	B POT
6	TRIGGER
7	+ 5v
8	GND
9	UNUSED

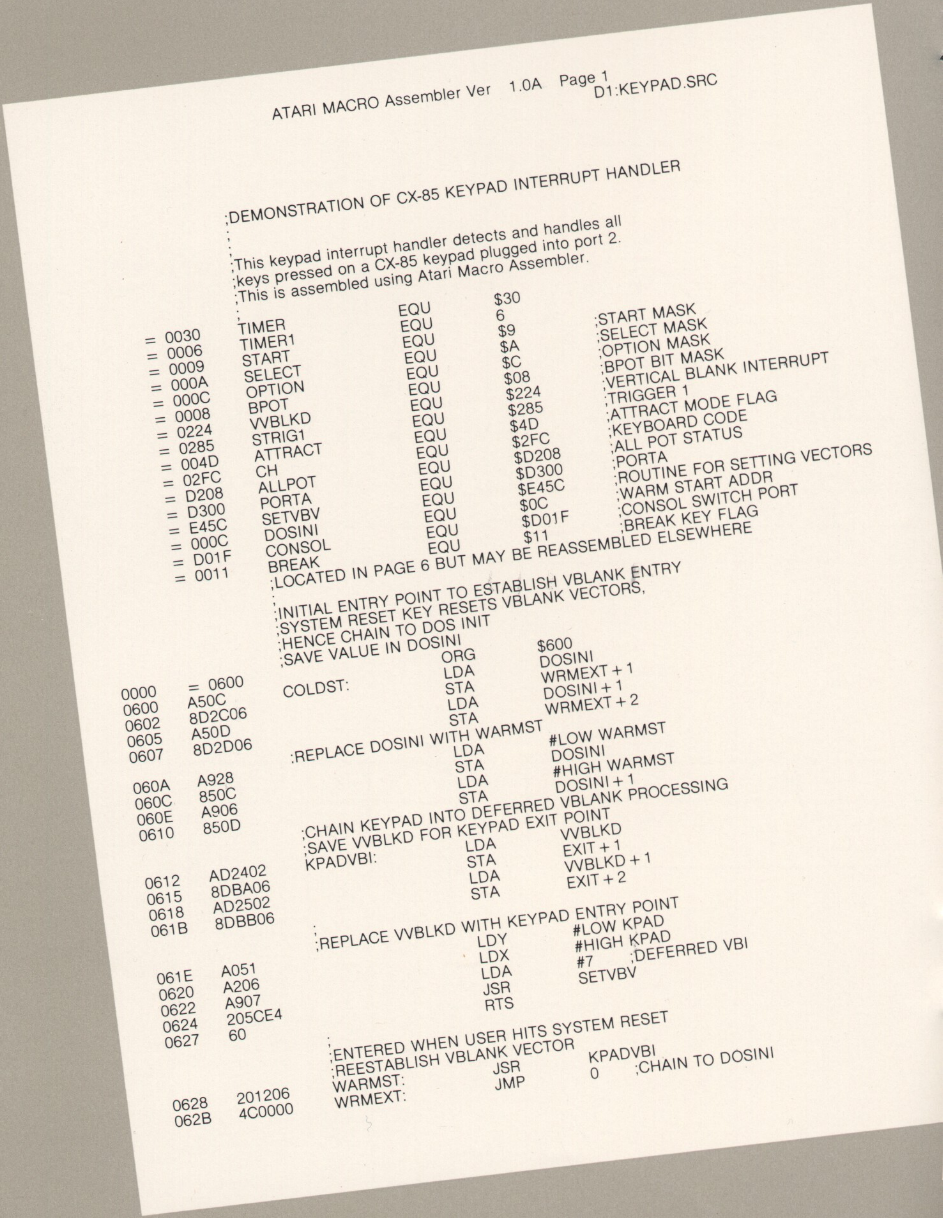
Figure 5



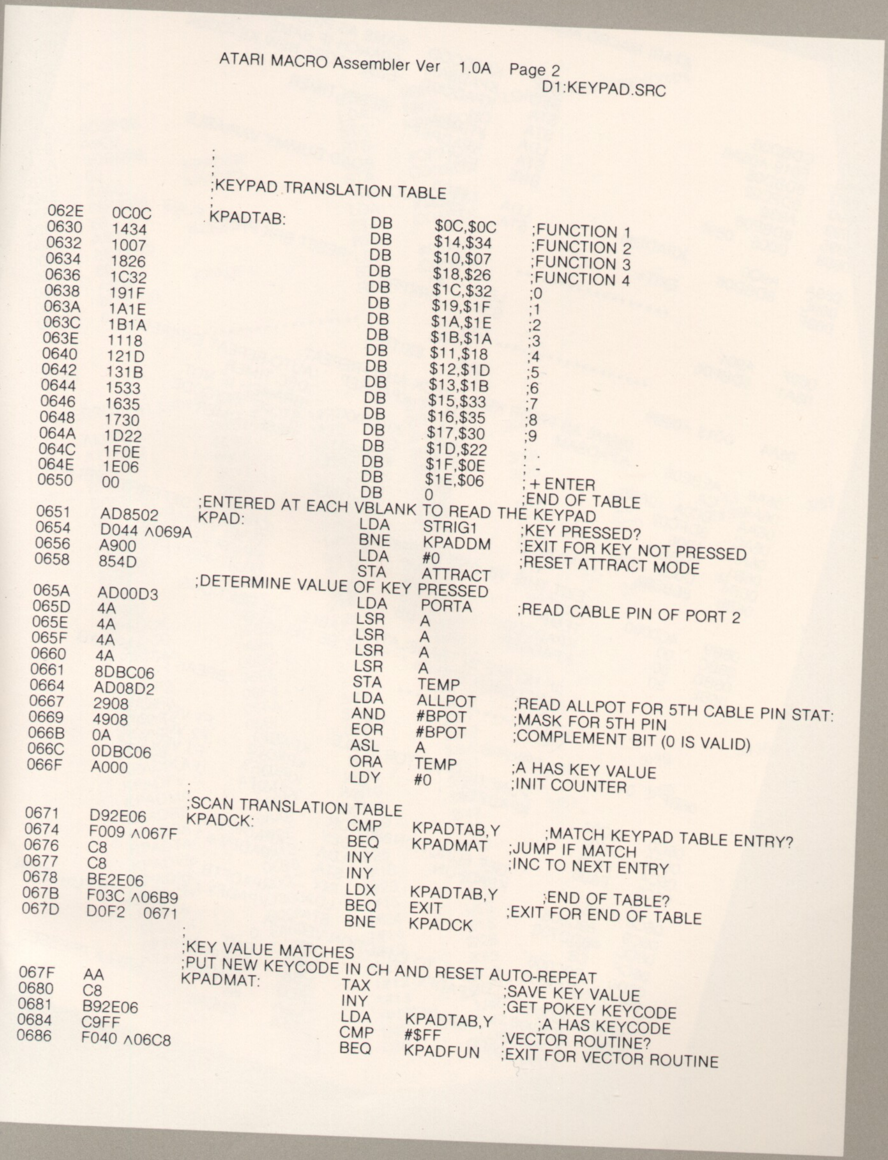
SIGNAL DEFINITION	
LOGIC "0"	0.4v AT 3.2 mA
LOGIC "1"	4.0v AT -0.7 mA

KEY	BPOT	STICK	HEX
0	0001	1100	$1C
1	0001	1001	$19
2	0001	1010	$1A
3	0001	1011	$1B
4	0001	0001	$11
5	0001	0010	$12
6	0001	0011	$13
7	0001	0101	$15
8	0001	0110	$16
9	0001	0111	$17
•	0001	1101	$1D
–	0001	1111	$1F
+ ENT	0001	1110	$1E
F1	0000	1100	$0C
F2	0001	0100	$14
F3	0001	0000	$10
F4	0001	1000	$18

Keypad Interrupt Handler Source Code

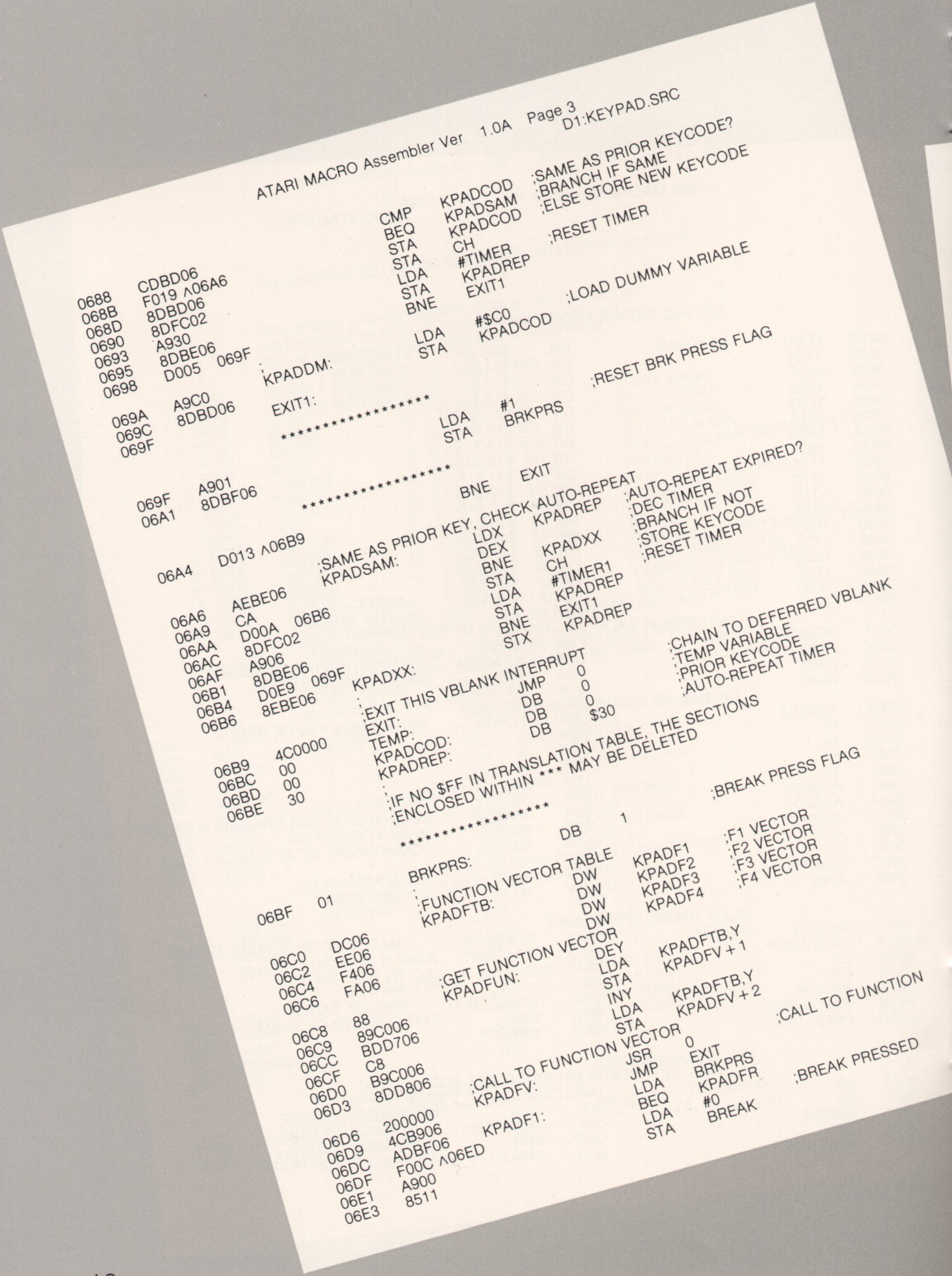


```
                      ATARI MACRO Assembler Ver   1.0A   Page 1
                                                          D1:KEYPAD.SRC

                 ;DEMONSTRATION OF CX-85 KEYPAD INTERRUPT HANDLER
                 ;
                 ;
                 ;This keypad interrupt handler detects and handles all
                 ;keys pressed on a CX-85 keypad plugged into port 2.
                 ;This is assembled using Atari Macro Assembler.
                 ;
   = 0030        TIMER          EQU       $30
   = 0006        TIMER1         EQU       6
   = 0009        START          EQU       $9         ;START MASK
   = 000A        SELECT         EQU       $A         ;SELECT MASK
   = 000C        OPTION         EQU       $C         ;OPTION MASK
   = 0008        BPOT           EQU       $08        ;BPOT BIT MASK
   = 0224        VVBLKD         EQU       $224       ;VERTICAL BLANK INTERRUPT
   = 0285        STRIG1         EQU       $285       ;TRIGGER 1
   = 004D        ATTRACT        EQU       $4D        ;ATTRACT MODE FLAG
   = 02FC        CH             EQU       $2FC       ;KEYBOARD CODE
   = D208        ALLPOT         EQU       $D208      ;ALL POT STATUS
   = D300        PORTA          EQU       $D300      ;PORTA
   = E45C        SETVBV         EQU       $E45C      ;ROUTINE FOR SETTING VECTORS
   = 000C        DOSINI         EQU       $0C        ;WARM START ADDR
   = D01F        CONSOL         EQU       $D01F      ;CONSOL SWITCH PORT
   = 0011        BREAK          EQU       $11        ;BREAK KEY FLAG
                 ;LOCATED IN PAGE 6 BUT MAY BE REASSEMBLED ELSEWHERE
                 ;
                 ;INITIAL ENTRY POINT TO ESTABLISH VBLANK ENTRY
                 ;SYSTEM RESET KEY RESETS VBLANK VECTORS,
                 ;HENCE CHAIN TO DOS INIT
                 ;SAVE VALUE IN DOSINI
0000    = 0600                  ORG       $600
0600    A50C     COLDST:        LDA       DOSINI
0602    8D2C06                  STA       WRMEXT + 1
0605    A50D                    LDA       DOSINI + 1
0607    8D2D06                  STA       WRMEXT + 2
                 ;REPLACE DOSINI WITH WARMST
060A    A928                    LDA       #LOW WARMST
060C    850C                    STA       DOSINI
060E    A906                    LDA       #HIGH WARMST
0610    850D                    STA       DOSINI + 1
                 ;CHAIN KEYPAD INTO DEFERRED VBLANK PROCESSING
                 ;SAVE VVBLKD FOR KEYPAD EXIT POINT
0612    AD2402   KPADVBI:       LDA       VVBLKD
0615    8DBA06                  STA       EXIT + 1
0618    AD2502                  LDA       VVBLKD + 1
061B    8DBB06                  STA       EXIT + 2
                 ;
                 ;REPLACE VVBLKD WITH KEYPAD ENTRY POINT
061E    A051                    LDY       #LOW KPAD
0620    A206                    LDX       #HIGH KPAD
0622    A907                    LDA       #7       ;DEFERRED VBI
0624    205CE4                  JSR       SETVBV
0627    60                      RTS
                 ;
                 ;ENTERED WHEN USER HITS SYSTEM RESET
                 ;REESTABLISH VBLANK VECTOR
0628    201206   WARMST:        JSR       KPADVBI
062B    4C0000   WRMEXT:        JMP       0        ;CHAIN TO DOSINI
```

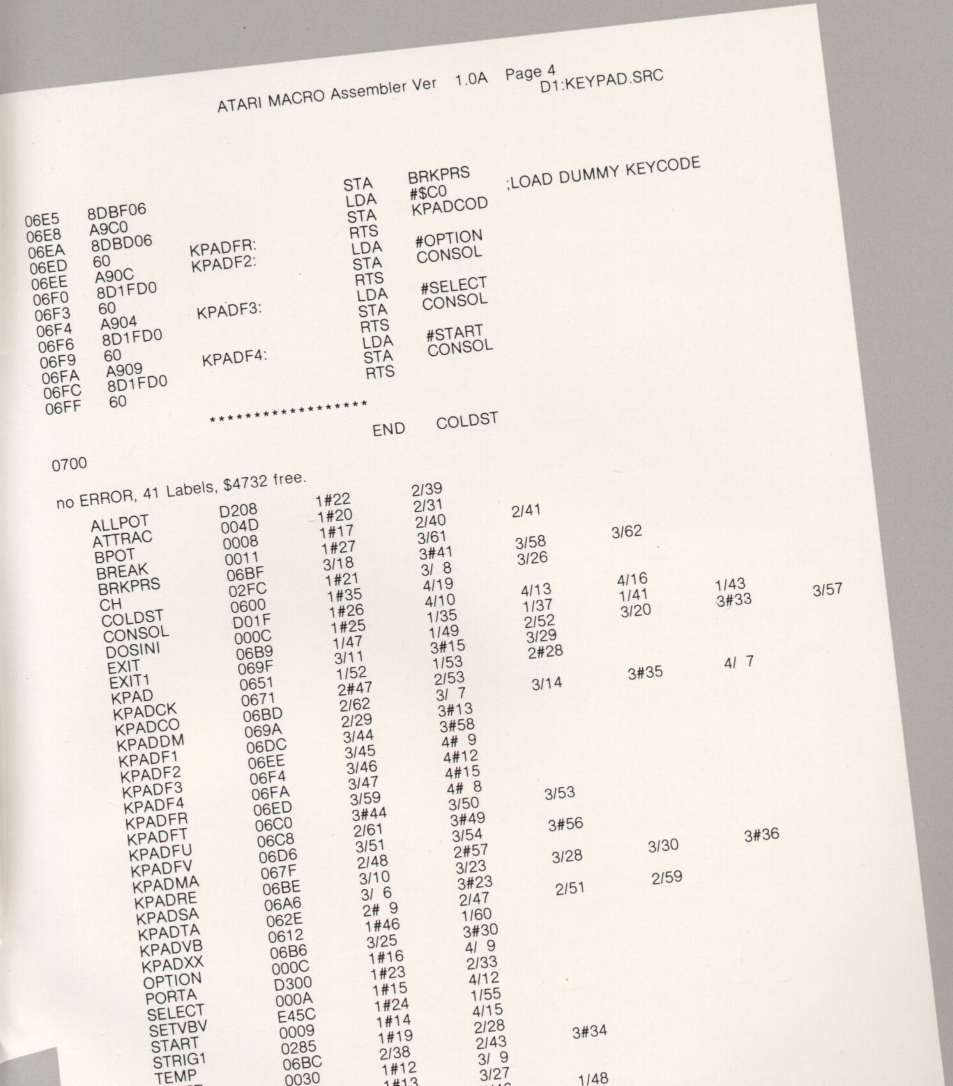


```
                    ATARI MACRO Assembler Ver   1.0A   Page 2
                                                       D1:KEYPAD.SRC

                   ;
                   ;
                   ;KEYPAD TRANSLATION TABLE
                   ;
062E   0C0C        KPADTAB:          DB     $0C,$0C     ;FUNCTION 1
0630   1434                          DB     $14,$34     ;FUNCTION 2
0632   1007                          DB     $10,$07     ;FUNCTION 3
0634   1826                          DB     $18,$26     ;FUNCTION 4
0636   1C32                          DB     $1C,$32     ;0
0638   191F                          DB     $19,$1F     ;1
063A   1A1E                          DB     $1A,$1E     ;2
063C   1B1A                          DB     $1B,$1A     ;3
063E   1118                          DB     $11,$18     ;4
0640   121D                          DB     $12,$1D     ;5
0642   131B                          DB     $13,$1B     ;6
0644   1533                          DB     $15,$33     ;7
0646   1635                          DB     $16,$35     ;8
0648   1730                          DB     $17,$30     ;9
064A   1D22                          DB     $1D,$22     ;.
064C   1F0E                          DB     $1F,$0E     ;-
064E   1E06                          DB     $1E,$06     ;+ ENTER
0650   00                            DB     0           ;END OF TABLE
                   ;ENTERED AT EACH VBLANK TO READ THE KEYPAD
0651   AD8502      KPAD:             LDA    STRIG1      ;KEY PRESSED?
0654   D044 ^069A                    BNE    KPADDM      ;EXIT FOR KEY NOT PRESSED
0656   A900                          LDA    #0          ;RESET ATTRACT MODE
0658   854D                          STA    ATTRACT
                   ;DETERMINE VALUE OF KEY PRESSED
065A   AD00D3                        LDA    PORTA       ;READ CABLE PIN OF PORT 2
065D   4A                            LSR    A
065E   4A                            LSR    A
065F   4A                            LSR    A
0660   4A                            LSR    A
0661   8DBC06                        STA    TEMP
0664   AD08D2                        LDA    ALLPOT      ;READ ALLPOT FOR 5TH CABLE PIN STAT:
0667   2908                          AND    #BPOT       ;MASK FOR 5TH PIN
0669   4908                          EOR    #BPOT       ;COMPLEMENT BIT (0 IS VALID)
066B   0A                            ASL    A
066C   0DBC06                        ORA    TEMP        ;A HAS KEY VALUE
066F   A000                          LDY    #0          ;INIT COUNTER
                   ;
                   ;SCAN TRANSLATION TABLE
0671   D92E06      KPADCK:           CMP    KPADTAB,Y       ;MATCH KEYPAD TABLE ENTRY?
0674   F009 ^067F                    BEQ    KPADMAT     ;JUMP IF MATCH
0676   C8                            INY                ;INC TO NEXT ENTRY
0677   C8                            INY
0678   BE2E06                        LDX    KPADTAB,Y       ;END OF TABLE?
067B   F03C ^06B9                    BEQ    EXIT        ;EXIT FOR END OF TABLE
067D   D0F2   0671                   BNE    KPADCK
                   ;
                   ;KEY VALUE MATCHES
                   ;PUT NEW KEYCODE IN CH AND RESET AUTO-REPEAT
067F   AA          KPADMAT:          TAX                ;SAVE KEY VALUE
0680   C8                            INY                ;GET POKEY KEYCODE
0681   B92E06                        LDA    KPADTAB,Y       ;A HAS KEYCODE
0684   C9FF                          CMP    #$FF        ;VECTOR ROUTINE?
0686   F040 ^06C8                    BEQ    KPADFUN     ;EXIT FOR VECTOR ROUTINE
```

Keypad Interrupt Handler Source Code



```
ATARI MACRO Assembler Ver  1.0A   Page 3
                                   D1:KEYPAD.SRC

0688    CDBD06                  CMP  KPADCOD    ;SAME AS PRIOR KEYCODE?
068B    F019 ^06A6              BEQ  KPADSAM    ;BRANCH IF SAME
068D    8DBD06                  STA  KPADCOD    ;ELSE STORE NEW KEYCODE
0690    8DFC02                  STA  CH
0693    A930                    LDA  #TIMER     ;RESET TIMER
0695    8DBE06                  STA  KPADREP
0698    D005   069F             BNE  EXIT1
                        ;
069A    A9C0            KPADDM: LDA  #$C0       ;LOAD DUMMY VARIABLE
069C    8DBD06                  STA  KPADCOD
069F                    EXIT1:
                        ;*****************
069F    A901                    LDA  #1         ;RESET BRK PRESS FLAG
06A1    8DBF06                  STA  BRKPRS
                        ;*****************
06A4    D013 ^06B9              BNE  EXIT
                        ;SAME AS PRIOR KEY, CHECK AUTO-REPEAT
06A6    AEBE06          KPADSAM:        LDX  KPADREP    ;AUTO-REPEAT EXPIRED?
06A9    CA                      DEX             ;DEC TIMER
06AA    D00A   06B6             BNE  KPADXX     ;BRANCH IF NOT
06AC    8DFC02                  STA  CH         ;STORE KEYCODE
06AF    A906                    LDA  #TIMER1    ;RESET TIMER
06B1    8DBE06                  STA  KPADREP
06B4    D0E9   069F             BNE  EXIT1
06B6    8EBE06          KPADXX: STX  KPADREP
                        ;
                        ;EXIT THIS VBLANK INTERRUPT
06B9    4C0000          EXIT:   JMP  0          ;CHAIN TO DEFERRED VBLANK
06BC    00              TEMP:   DB   0          ;TEMP VARIABLE
06BD    00              KPADCOD:        DB   0  ;PRIOR KEYCODE
06BE    30              KPADREP:        DB   $30        ;AUTO-REPEAT TIMER
                        ;
                        ;IF NO $FF IN TRANSLATION TABLE, THE SECTIONS
                        ;ENCLOSED WITHIN *** MAY BE DELETED
                        ;*****************
06BF    01              BRKPRS: DB   1          ;BREAK PRESS FLAG
                        ;
                        ;FUNCTION VECTOR TABLE
06C0    DC06            KPADFTB:        DW   KPADF1     ;F1 VECTOR
06C2    EE06                    DW   KPADF2     ;F2 VECTOR
06C4    F406                    DW   KPADF3     ;F3 VECTOR
06C6    FA06                    DW   KPADF4     ;F4 VECTOR
                        ;GET FUNCTION VECTOR
06C8    88              KPADFUN:        DEY
06C9    89C006                  LDA  KPADFTB,Y
06CC    BDD706                  STA  KPADFV + 1
06CF    C8                      INY
06D0    B9C006                  LDA  KPADFTB,Y
06D3    8DD806                  STA  KPADFV + 2
                        ;CALL TO FUNCTION VECTOR
06D6    200000          KPADFV: JSR  0          ;CALL TO FUNCTION
06D9    4CB906                  JMP  EXIT
06DC    ADBF06          KPADF1: LDA  BRKPRS
06DF    F00C ^06ED              BEQ  KPADFR
06E1    A900                    LDA  #0         ;BREAK PRESSED
06E3    8511                    STA  BREAK
```



```
                    ATARI MACRO Assembler Ver   1.0A   Page 4
                                                  D1:KEYPAD.SRC

06E5   8DBF06                          STA    BRKPRS
06E8   A9C0                            LDA    #$C0       ;LOAD DUMMY KEYCODE
06EA   8DBD06                          STA    KPADCOD
06ED   60           KPADFR:            RTS
06EE   A90C         KPADF2:            LDA    #OPTION
06F0   8D1FD0                          STA    CONSOL
06F3   60                              RTS
06F4   A904         KPADF3:            LDA    #SELECT
06F6   8D1FD0                          STA    CONSOL
06F9   60                              RTS
06FA   A909         KPADF4:            LDA    #START
06FC   8D1FD0                          STA    CONSOL
06FF   60                              RTS
                    ******************
0700                                   END    COLDST

no ERROR, 41 Labels, $4732 free.
     ALLPOT         D208       1#22       2/39
     ATTRAC         004D       1#20       2/31
     BPOT           0008       1#17       2/40      2/41
     BREAK          0011       1#27       3/61
     BRKPRS         06BF       3/18       3#41      3/58      3/62
     CH             02FC       1#21       3/ 8      3/26
     COLDST         0600       1#35       4/19
     CONSOL         D01F       1#26       4/10      4/13      4/16
     DOSINI         000C       1#25       1/35      1/37      1/41      1/43
     EXIT           06B9       1/47       1/49      2/52      3/20      3#33      3/57
     EXIT1          069F       3/11       3#15      3/29
     KPAD           0651       1/52       1/53      2#28
     KPADCK         0671       2#47       2/53
     KPADCO         06BD       2/62       3/ 7      3/14      3#35      4/ 7
     KPADDM         069A       2/29       3#13
     KPADF1         06DC       3/44       3#58
     KPADF2         06EE       3/45       4# 9
     KPADF3         06F4       3/46       4#12
     KPADF4         06FA       3/47       4#15
     KPADFR         06ED       3/59       4# 8
     KPADFT         06C0       3#44       3/50      3/53
     KPADFU         06C8       2/61       3#49
     KPADFV         06D6       3/51       3/54      3#56
     KPADMA         067F       2/48       2#57
     KPADRE         06BE       3/10       3/23      3/28      3/30      3#36
     KPADSA         06A6       3/ 6       3#23
     KPADTA         062E       2# 9       2/47      2/51      2/59
     KPADVB         0612       1#46       1/60
     KPADXX         06B6       3/25       3#30
     OPTION         000C       1#16       4/ 9
     PORTA          D300       1#23       2/33
     SELECT         000A       1#15       4/12
     SETVBV         E45C       1#24       1/55
     START          0009       1#14       4/15
     STRIG1         0285       1#19       2/28
     TEMP           06BC       2/38       2/43      3#34
     TIMER          0030       1#12       3/ 9
     TIMER1         0006       1#13       3/27
     VVBLKD         0224       1#18       1/46      1/48
     WARMST         0628       1/40       1/42      1#60
     WRMEXT         062B       1/36       1/38      1#61
```

ATARI®

A Warner Communications Company

Every effort has been made to ensure the accuracy of the documentation in this manual. However, because ATARI, INC. is constantly improving and updating the computer software and hardware, we are unable to guarantee the accuracy of the printed material after the date of publication and disclaim liability for changes, errors or omissions.

PRINTED IN U.S.A. **CO61037 REV. A**