CHIPOS SUBROUTINES (& Calling Sequences)

ERASE = C079

Function: Clears the display buffer.
Affected parameters: A, X, DISBUF
Calling sequence:

```
BD C079      JSR  ERASE
```

FILL = C07D

Fill part or all of display buffer with constant byte.
Input: A = byte to show
X = starting location; e.g. 0100 for whole screen
Affected: X, DISBUF (stops at 0200)
Call:

```
CE xxxx      LDX  #$xxxx    start locn
86 kk        LDA A  #$kk    load constant
BD C07D      JSR   FILL
```

RANDOM = C132

Generate a pseudorandom byte.
Input: (optional: initialize RNDX+1 & RND)
Output: A, RND
Affected: RNDX, X
Call:

```
BD C132      JSR    RANDOM
```

LETDSP = C193

Called prior to SHOWI to display a hex digit as a 3x5 symbol.
Input: A = digit to be displayed
Output: I, DDPAT
Affected: A, B, X, PATNH, PATNL
Call:

```
                 LDA A  xxx     load digit
BD C193          JSR    LETDSP  setup I and DDPAT
C6 05            LDA B  #5      show 5-byte pattern
BD C224          JSR    SHOWI
```

DECEQ = C1E0

Store 3-digit BCD equivalent of A at X, X+1, X+2.
Input: A = unsigned binary no.
Output: Memory at X, X+1, X+2 (3 bytes)
Affected: A, B, X (=X+3)
Call:

```
                 LDA A  xxx     byte to convert
                 LDX            loc'n for result
BD C1E0          JSR    DECEQ
```

SHOWI = C224, SHOWX = C226

SHOWI (-X) displays an N-byte symbol in memory pointed at by I (X).
Display dots are XORed with existing dots.
Horizontal and vertical wrap-round occurs across borders.
If N = 0 (B-reg.), then N = 16 is assumed.
Input: I (or X) = pointer; B = N (no. of bytes);
VX, VY = screen coordinates for pattern.
Output: DISBUF, VF (=01 if overlap)
Affected: A, B, X (=X+N), VY (=VY+N), VF, BLOC, ZHI, PATNH-L
Call: Initialize: VX, VY, VF=00 (option), I or X, B (=N), then:-

```
BD C224/6    JSR    SHOWI/X
```

DISLOC = C275

Computes the address of the display byte at coords (B, VY).
Input: BLOC = $01 (DISBUF MSB),
B = x-coordinate; VY = y-coord.
Output: X, BLOC+1 (adrs of req'd byte)
Affected: A, B.
Note: The dot position within the byte is determined from VX, (LS 3 bits).

PAINZ = C287

Initializes the keypad port; clears PIA flags; disables CA1 IRQ.
Affected: B, X (=PIA adrs), PIAA, PIAA+1

KEYINP = C297

Decodes the hex keypad, after a de-bounce delay of 3.33 msec.
A flag, BADRED, is set to $0F if a bad read occurred, or if no key was down, else BADRED = 00.
Returned: A = hex keycode; KEYCOD (=A); BADRED.
Affected: A, B, X (= #PIAA).

GETKEY = C2C4

Waits for a key to be pressed, then, if it's a HEX key, calls KEYINP; if FN key, returns with A = $8C (negative). A valid keystrike is acknowledged with a BLEEP.
Returned: A = hex or 8C (FN). (See also KEYINP, BLEEP, RTC.)
Affected: B, KEYCOD, BADRED, TONE, XTEMP (X is saved)

BLEEP = C2DF

Generates a 2400 Hz tone in speaker for approx 80 msec.
Affected: B, TONE

BTON = C2E5

Generate a variable length tone at either 2400 Hz or 1200 Hz.
Enter: B = $40 (1200Hz) or $41 (2400Hz)
TONE = duration x 20 msec (eg: TONE=05 for 100ms)

DEL333 = C2F3; (DEL167 = C2F5)

Delay for 3.33 msec (1.67 msec), assuming display/DMA is off.
Affected: nil.

PBINZ = C2FE

Initialize: serial I/O (tape), tone, RTC timer and display/DMA.
Input: A = $36 RTC off, DMA off
A = $3E RTC off, DMA on
A = $37 RTC on, DMA off
A = $3F RTC on, DMA on
Affected: B, X (= #PIAB), PIAB, PIAB+1

INBYT = C310

Inputs a byte from the serial data input line PB7, in standard 300 Baud async format (1 start bit, 8 data bits, 1 or more stop).
Note: Display/DMA must be disabled (see PBINZ, above).
Affected: A (returned byte), B, XTEMP.

OUTBYT = C32B

Outputs a byte (A) to the serial data output line PB0 at 300 Bd.
Note: Display/DMA must be disabled first.
Affected: B, XTEMP. (A and X are preserved)

BYTIN = C390

Accepts 2 hex digits from the keypad and builds a composite byte (A).
Returned: A.
Affected: A, B, ATEMP, (see also: GETKEY).

START = C360

Monitor entry point.
Terminate a machine-code program with: JMP $C360.

SHODAT = C3C8

Displays a byte (2 hex digits) in memory, pointed at by X, on the bottom part of the screen.
Enter: X = loc'n of byte to show;
VX = horiz. cursor position.
Note: VY must be set to $1A prior to the first call to SHODAT.
The display buffer from $01C8 onwards must be cleared also.

SHOBYT = C3CA

Same as SHODAT except the byte to be shown is in the A-reg.
Affected: A, B, VX (=VX+8); same for SHODAT.

DIGOUT = C3D2

Similar to SHOBYT, but only one digit is displayed (LS 4 bits).

CURSR = C3DC

Used in conjunction with SHODAT, SHOBYT & DIGOUT. Moves "cursor" position to the right 4 dots; ie. adds 4 to VX.
Enter: nil.
Affected: A, VX (=VX+4), VY (=$1A), X (=XTEMP)

CURS1 = C3E0

Used with SHODAT, SHOBYT, DIGOUT, to reset "cursor" position.
Enter: A = horiz. cursor position ($00 to $3C).
Affected: VX (=A), VY (=$1A), X (=XTEMP).
Note: The above routines will only work on the bottom row of the screen, i.e. VY is fixed at $1A.

Scratchpad parameter addresses for the above subroutines:

DDPAT	0008	RND	000D	ATEMP	000F
XTEMP	0012	ZHI	0014	ZLO	0015
KEYCOD	0017	BADRED	0018	BLOC	001C
PATNH	001E	PATNL	001F	TIME	0020
TONE	0021	I	0026	RNDX	002C
VX	002E	VY	002F	VF	003F
DISBUF	0100	PIAA	8010	PIAB	8012
