Screen-64 source docs (based on source from Feb 1, 1983)
This provides a 64x32 screen (4 x 6 pixel characters)

(C) 1983 by Intellectronics
Written by Lou Fiorino (original older "Z" code by Lou Firorino & Harvey Brofman)
Older source also shows scroll speed by POKE instead of CHR$:
POKE &H7EF7,0 = Slow scroll
POKE &H7EF7,>0 = Fast scroll

NOTE: I think the source code might be a bit older than the actual release. In particular, the "window" POKE's don't seem to work, although everything else does.
-LCB 04/07/2023-

4 versions included: S16, S16X, S32, S32X
S32 explicitly sets up SAM/VDG registers
S32X does not
S16 explicitly sets up SAM/VDG registers
S32

(I have had more success with the X versions)

Documentation from within source code starts here:

S16 says to:
```
 CLEAR200,&H6200
 LOADM "SCREEN64"
 EXEC
```

S16X says to:
```
 CLEAR200,&H3A00
 PMODE 4:LOADM "SCREEN64"
 EXEC
```

S32 says to:
```
 CLEAR200,&H6200
 LOADM "SCREEN64"
 EXEC
```

S32X says to:
```
 CLEAR200,&H7A00
 PMODE 4:LOADM "SCREEN64"
 EXEC
```

Control codes (and how to make a window):
CHR$() code  Function

```
1                   Screen Color  (swaps between green and white color sets)
2                   Home Cursor
3                   Invert Screen (leaves text/background colors unchanged)
4                   Full Screen
5                   Scroll Speed  (toggles smooth scroll/regular scroll)
7                   Bell
8                   Backspace
9                   Tab           (8 chars per tab spot)
10                  Line Feed
12                  Clear Screen
13                  Carriage Return
20                  Reverse On/Off
21                  Cursor Up
22                  Cursor Down
23                  Cursor Back
128                 SetXY (next 2 characters are X & Y)

To form a window simply:
  POKE &H7FE0,FIRST LINE (0-31)
  POKE &H7FE1 LAST LINE (0-31)
Last line must come *after* the first line.

(also &H7FE2 is the fast scroll flag, but you can do that through CHR$(5))
```