Here are details of the patches and changes made to the various binaries

**Tiny Basic**

**TinyBasic\_$2000\_raw.kpt**

A straight dump of a copy of Tiny Basic loaded from a cassette tape I own with a 1978 copyright by Tom Pittman. It loads from $2000-$28FF. Cold start is at $2000 and warm start is at $2003. On cold start it sets up a lot of variables in page zero. It also has the Break function for the KIM-1 already embedded at $28DO. The default setting sends excess pad characters and x-on and x-off characters that can be confusing on a modern terminal emulator.

**TinyBasic\_$2000\_VT100.kpt**

This is a version of the above that I patched to work more smoothly with the ANSI (VT100) emulation in RealTerm. I needed to find room for a little additional code. When examining the Break function at $28D0 I found a code fragment immediately following the Break function that accessed locations at $6e02 and a subroutine at 72C6. This looked like code to wait for a bit change at 6E02 and load either $01 or $FF into the accumulator before call a subroutine at $72C6. Since none of these addresses are typical of KIM expansions and I could find no JMP or JSR to the address of this code, I used the space starting at $28E8 to implement a control character filter to remove the unnecessary x-on, x-off and rubout characters.

Patches:

$200A $E8 – Low byte of character output routine with filter to remove control characters

$200B $28 - High byte of character output routine with filter to remove control characters

$200F $08 – replaced $5F (\_) with Backspace to work with VT100 emulation

$2011 $80 – selects $FF as the pad character and # pad characters to 0. Note TB still outputs one pad character after the : prompt even when the pad character count is set to zero.

Function to strip control characters from output stream. Described in Tiny Basic manual appendix C.

28E8 C9 00 CMP #$00 ;Ensure flags are current for accum just in case

28EA 10 01 BPL .+1 ;If bit 7 = 0 skip next instruction

28EC 60 RTS ;Return with no action

28ED 4C A0 1E JMP OUTCH ;Jump to KIM char output function

**Microsoft Basic for KIM-1**

**Kb9.txt**

There are several binaries and tape dumps for Kim Basic (KB9) floating around. The only one I could find that would run cleanly was from the Corshamtech.com website (buried in a Tech Tips page <https://www.corshamtech.com/tech-tips/basic-interpreters-for-the-6502/> ). The file was a paper tape dump which loaded fine except for a checksum error on the end record. This file strips all the extra nulls and corrects the EOR checksum error so it loads cleanly without ending with an error.

Cold start for KB9 is $4065. It starts and asks:

MEMORY SIZE? – enter upper memory limit in DECIMAL. Or hit Return and it will find the top of memory

TERMINAL WIDTH? - Return defaults to 72. I typically use 80

WANT SIN-COS-TAN-ATN? – Y = yes, recommended.

Warm start is a 0000. After a cold start KB9 makes changes in memory, especially if you answer ‘Y’ for the trig functions. It sets up a bunch of stuff in page zero including the warm start vector, so creating a functioning memory dump from KB9 after a cold start could be a challenge, especially if your memory space is not identical. This version and the following version load and cold start successfully on several systems with expansion memory starting at $2000 and ending at $7FFF or beyond.

**kb9\_VT100.txt**

The original KB9 was used “\_” (underscore $5F) as a backspace character. This makes it awkward to use with modern terminal emulators. I found a patch for this getting backspace ($08) to work correctly in an old Dutch KIM Users group newsletter by modifying the getline function.

<http://retro.hansotten.nl/uploads/kimkenner/kimkenner05.pdf>

The changes to just 6 bytes are highlighted in bold and underlined. These are the only changes incorporated in kb9\_VT100.kpt to create a clean loadable copy that works well with VT-100 emulators.

2420 ca br2420 dex

2421 10 05 bpl br2438

2423 20 bf 29 br2423 jsr crlf

2436 a2 00 gelin ldx #$00

2428 20 56 42 br2428 jsr getch from kim

242b c9 07 cmp #$07 ;bell is a valid char

242d f0 14 beq br2443

242f c9 0d cmp #0d ; carriage return?

2431 f0 20 beq br2420

2433 c9 **08**  cmp #$08 ;rubout?

2435 **f0** **e9** beq br2420 ;yes, then skip previous char

2437 c9 7d cmp #$7d ;char $7d, then skip it

2439 b0 ed bcs br2428

243b c9 40 cmp #$40 ;cancel line?

243d f0 e4 beq br2423

243f c9 **20** cmp #$20 ;char < 20, then skip it

2441 **90** **e5** bcc br2428

**FOCAL**

**focal\_FCL65E\_raw.txt**

A straight dump of a cassette load of FCL65E copyright 1978 Wayne Wall. The tape was label with:

ID:01 $0020-00D3

ID:02 $2000-35F3

This file combines these two segments and has an EOR corrected for the total number of lines. This straight from the factory dump starts at $2000 and has the upper memory limit $0053-$0054 set to $3F00. It loads and runs in just an 8K memory expansion with a cold start at $2000. Rubout ($7F) is the delete character that generates a ‘\’.

**focal\_FCL65E\_VT100.txt**

As with most early code, Focal was set up to use a teletype and has extra pad characters. The following patches have been made for video terminal operation:

34B3 EA

34B4 EA

34B5 EA

34B6 EA

34BA 18

34BB 60

34C4 29

34C5 7F

34C6 18

34C7 60

Since my system has RAM to $7FFF, I made the following changes location $0053 = 00, $0054 =7F. If your memory space is different, you may need to modify these locations in the file. In VT100 emulation, backspace appears to work but this is not fully confirmed. Delete ($7F) does work generating a ‘\’ with each key press.