Last Updated: December 21, 2018

VIP2K Monitor Program Ver. 1.4

NOTE: RCA Basic 3 by Ron Cenker Copyright 1981
NOTE: The VIP2K Membership Card Kit by Lee Hart, Copyright 2018
NOTE: VIP2K Monitor Program by Chuck Yakym Copyright 2018
NOTE: Modified Chip 8 by Marcel van Tongeren

NOTE: This monitor contains a scroll variable for the VIP2K display screen. It is stored in memory location FEF8h. By default this location contains a value of 00h which means scroll is turned on, if you write any non-zero value into this location it will turn the scroll off. Note: When you run Basic 3 and return back to the monitor program, the scroll will be turned back on.

The ESCAPE (ESC) key is used as a "BREAK" key and will stop processing your command when inputted into the command line. For example: a W8000 11 22 33<ESC> will only WRITE the data 11 and 22 into location starting at address 8000, the 33 data is ignored and not written into memory. This ESCAPE function is active on ALL commands that you input. When you use the ESCAPE key the last input value is ignored and will result in displaying a "Function Aborted" message.

NOTE: ESCAPE is not always being checked for when the Monitor is performing a selected function,

When using the "M" (read memory function) a input command of M1000 0100<CR> will display 100 Hex (or 256 decimal) bytes of memory location starting at 1000 thru 10FF. The input command of M1000 80<CR> will display the contents of memory location 1000 thru 107F. If the inputted value of the number of bytes to display equals zero then this function will be aborted.

NOTE: If you want to abort the listing command while it is displaying the memory contents you MUST press the "ESC Key”.

When using the "W" (write memory function) an input of W1000<CR> will result in a "Function Aborted" (No data to write). a W8000 11<CR> will write a value of 11 into location 1000. Now a W8000 11 22 33<ESC>, this will only write the 11 data into location 8000 and the data 22 into location 8001. The last 33 is ignored via the ESC key being pressed.

When using the "T" (transfer memory) command it requires three values.For example the command T8000 9000 0100<CR>, will transfer the memory from 8000 to 9000 for 0100 Hex bytes. If the inputted value of the number of bytes to transfer equals zero then this function will be aborted.

When using the "R" (RUN) command it will transfer control over to your program. For example, R8000<CR> starts executing the code loaded into location 8000 with X =2 and the program counter is set to R3, which would contain the start address of 8000. The CDP1802 registers will be preloaded with the values displayed using the “V”iew register command with the exception of R3. Register values can be changed by using the “W”rite command. (A description of this follows below.)

Run Commands precautions:
Never use or change the value of R0, and R1. These are used for the Interrupt routine and DMA OUT.
Never change your routine Program counter to either R4 or R5.
Always use R2 as your stack pointer, when the Run command jumps to your routine R2 is already pointing to valid memory for use as a stack. If you change the value of R2 within your routine then make sure that it points to within the User RAM area of memory. See VIP2K Memory Map.

To back to the VIP2K Monitor from your routine, all you have to do is a jump to address 0D1B.
(C0 0D 1B)

Changing registers using the “W”rite command prior to using the “R”un command:

To change the value of a register, all you have to do is to write the value into the following memory locations.

Address	Register
E7FF	D
E7FE	T
E7FD	X
E7FC	P
E7DB	DF
E7DC-E7DD	R0
E7DE-E7DF	R1
E7E0-E7E1	R2
E7E2-E7E3	R3
E7E4-E7E5	R4
E7E6-E7E7	R5
E7E8-E7E9	R6
E7EA-E7EB	R7
E7EC-E7ED	R8
E7EE-E7EF	R9
E7F0-E7F1	RA
E7F2-E7F3	RB
E7F4-E7F5	RC
E7F6-E7F7	RD
E7F8-E7F9	RE
E7FA-E7FB	RF

HIGH-LOW register

When using the "V" (view) command, remember that it only displays the random contents of the memory where they are stored in the scratchpad area at the time of entering this Monitor program. If this Monitor is re-entered via your program called by the "R" command it will display the correct values of all registers. See the “R”un command for more information.

IMPORTANT NOTE: When using the “L” (LOAD) or “S” (SAVE) command, make sure that your PC terminal program is already to send or receive serial data. By default, this monitor program uses a serial baud rate of 9600. However it can be changed by the user.

Changing baud rates:
This is done by changing the baud rate timing bytes at memory location E7CDh and E7CEh.

Memory		
E7CD	E7CE	Baud Rate
01	0d	9600
01	19	4800
01	35	2400
01	68	1200

These values can be written by using the “W” (WRITE) command in the monitor or by using the POKE command in Basic 3. NOTE: When you return from Basic 3 by using its BYE command then the baud rate will be reset back to its default value of 9600 baud.

When using the "L" (LOAD) user program command, It will promptly display the following message "Ready to LOAD program" then the screen is turned off during the transfer. IF you decide not to load a program, you can just press the <ESC> Escape key on your PC terminal to abort this function. Once the file transfer has begun, you can NOT abort this function. You use the "Send" or “ASCII Upload” command of your terminal program to load the user program in with. After your file has been loaded in, the screen will be then turned back on and this monitor should display "File Loaded Successfully" or display any error that may have been detected. Only files in I8HEX file format are understood by this command.

NOTE: If you plan on “loading” Intel Hex I8HEX format files, then please do the following,
Under the “Send” tab in the Realterm terminal program, set to “Delays” as follows:
For a 9600 baud rate, it "MUST" be equal to or greater that a value of 5.
For a 4800 baud rate, it "MUST" be equal to or greater that a value of 10.
For a 1200 or 2400 baud rate, it “MUST” be equal to or greater than a value of 15.
For a 300 baud rate, it “MUST” be equal to or greater than a value of 50.
Any lower “Delays” values may cause the Monitor Program to crash or not load the file correctly If that happens, then you may have to restart the Monitor Program. The above values are where I tested the load functions at, however due to the many factors that can effect these values, you may have to change them so they match your computer to Membership Card timing requirements needed for the load function.
NOTE: These delay values aren’t needed for the save function.

Note: When saving a program, make sure that your PC terminal program is all set up and ready to start capturing the serial data that will be sent to it.

When using the "S" (SAVE) user program command, it will promptly display the following message "Ready to SAVE program" then the screen will be turned off and the transfer of data will begin immediately. You can NOT abort this function. You use the "Capture" command of your terminal program to save the user program with. After your file has been saved the screen is turned back on and , this monitor should display "File Saved Successfully".

The format for the Save command is "Saaaa bbbb<CR>, where aaaa is the starting address to save from and bbbb is for how many bytes followed by a <CR> Return key. The data transmitted to your terminal program is in I8HEX file format.

When using the "D" (disassembler function) the outputted display will be in the following format: AAAA O M X, where "AAAA" is the address being disassembled, "O" is the complete command Opcode, "M" is the Mnemonic, "X" is any other information needed to complete the disassembly of the Opcode. For example, if you disassemble the following RCA opcodes 7B, F8, 10, A8 from starting address84000h, the displayed output will be as follows:

NOTE: IF your ending address is greater than 32K higher than your starting address, the disassembler will only disassemble the starting address opcode and then return you to the monitor’s main command prompt.

```
8000 7B     SEQ
8001 F810   LDI 10
8003 A8     PLO R8
```

The Format for using the "D" disassembler command is Daaaa bbbb<CR>
Where aaaa is your starting address and bbbb is the ending address
If the inputted ending address is equal to or less than your starting address OR your ending address is 0000h the Disassembler will enter into the Single Step Mode.

Disassembler Modes:

Auto/Normal Mode:

This will automatically display the disassembled results from the inputted starting address to the ending address inputted. On a long listing you can abort by press the ESC key.

Single Step Mode:

There are three ways to automatically enter the Single Step Mode. It is entered by entering an ending address that is either equal to your starting address, less than your starting address, or an ending address that is 0000h. Another easy method to invoke single step mode is to type "8000 0 " to debug from address 8000. (Note the space following the last zero).This Mode will display the disassembled output one opcode at a time and wait for a user input (<CR>) to continue. After the disassembly is displayed the disassembler is waiting to see If you hit the <ESC> Escape key the disassembler is aborted and returns you back to the monitor's main command prompt or seeing if you hit the <CR> key depression that will continue to disassemble the next opcode. To keep the display "clean" looking, your key depressions are not echoed back to the terminal screen while in the Single Step Mode.

NOTE: If you want to personally test the Disassembler and see all the possible mnemonics that it will generate, then do the following command from the Monitor's command prompt. D7700 7826<CR>

RCA BASIC3

To enter RCA Basic3, enter the "B" command. For instructions on how to use Basic3 please refer to the owner's manual file named Basic3v11user.pdf

Note: The PSAVE and PLOAD commands work just like the monitor's Load and Save command except that Basic 3 once you start the PSAVE or PLOAD command you can't abort from it. Please make sure that your PC terminal program is configured correctly and is ready for the serial I/O.

The ESC key is used also as Basic 3's Break key.

Chip 8

The "C"hip 8 command runs an already loaded Chip 8 program. If you run the "C" command and there isn't a Chip 8 file loaded into memory already, the monitor will tell you "No Chip 8 File Loaded" then return you back to the monitor prompt.

The "I" command is used to load a Chip 8 program file into memory. This file must be in I8HEX form in order to be loaded in correctly. If you try to load a "binary" Chip 8 program with this command, then the monitor will crash and then you will need to reset your VIP2K system.

NOTE: This monitor contains a Chip 8 variable and is located at FEFDh. This memory contains either 00h (default value) or 80h depending on whether a Chip 8 program has been loaded into memory. A value of 00h indicates that there is no Chip 8 program loaded, however if a Chip 8 program has been loaded then it contains a value of 80h. If you load and run a Chip 8 program then exit back to the monitor then this variable is reset back to 00h, however if you want to rerun your Chip 8 program, assuming that you have not modified any memory that is used by your Chip 8 program, you can re-enter your Chip 8 program without reloading it. To do this, simply use to the monitor to write to memory address FEFDh with an 80h (WFEFD 80<cr>), then use the "C"hip 8 command to run your program again.

Chip 8 Interpreter Information provided by Marcel van Tongeren for use on the VIP2K system

Chip 8 Interpreter

The VIP2K Chip 8 Interpreter is designed by Marcel van Tongeren and was based on the VIP Chip 8 Interpreter.

Any chip 8 software should run on the VIP2K but due to the larger screen resolution and faster CPU some speed differences will be noticeable. There are also some differences in the Chip 8 instruction set but besides missing sound support should not be noticeable. Chip 8X, Chip 10, Chip ETI600, Chip 8 Hires and Chip 8 using 1802 subroutines are not supported.

Before starting Chip 8 with the 'C' command make sure to load Chip 8 SW in Intel hex format with the 'I'' command.

Key Functions

During runtime the following key functions are supported:
Control Q: Quit chip 8
Control 1 to 9: Speed where 1 is slowest, 9 is fastest
Control K: Reset keyboard map to default
Control R: Reset Chip 8 interpreter
Control S: Reset Chip 8 interpreter and start-up in small screen mode, i.e. 2x4 pixel size.
Control L: Reset Chip 8 interpreter and start-up in large screen mode, i.e. 3x5 pixel size

The smaller screen size is recommended for software that uses more and larger graphic or screen updates.

Default Key Mapping

0-9, A-F: Chip 8 keys 0-F
I: Chip 8 key 2
J: Chip 8 key 4
K: Chip 8 key 6
M: Chip 8 key 8
Space: Chip 8 key 5

The definition of I, J, K, M and space allow easier game movement for most games I: up, J: left, K: right, M: down, space: fire.

Dedicated key definition can be included in the Chip 8 hex file by including the complete keyboard mapping table stored in RAM at 0xFF00-0xFFAF.

Location of game keys is at I: 0xFF11, J: 0xFF0A, K: 0xFF12, M:0xFF0B, space: 0xFF1B. In the same way default speed value (0xFFA8) and default screen size (0xFFA9) can be included.

Memory Map

8000-81FF: Normally not used but in theory could be used for Chip 8 code
8200-8FFF: Chip user space
E800-EC77: Video RAM

E800-E84D: Top 3 lines, not used by chip 8
E84E-EB8D: Chip 8 video screen for 3x5 pixel format
EB8E-EBF5: Bottom 4 lines normally used by chip 8, some games like Pong use some of these lines
EBF6-EC77: Bottom and top lines should always be 0

E800-E8CF: Top 8 lines, not used by chip 8
E8D0-EC0F: Chip 8 video screen for 2x4 pixel format
EC10-ECF9: Bottom 9 lines normally used by chip 8, some games like Pong use some of these lines
ECFA-ED95: Bottom and top lines should always be 0

FF00-FF9F: Key Mapping table
FFA0-FFA4: CHIP8 identifier text
FFA8: Speed, 0-0x30 in steps of 6
FFA9: Screen resolution, 0: large with 3x5 pixel size, NOT 0: small with 2x4 pixel size
FFB2-FFCF: Jump table for Chip 8 instructions

FFE0-FFEF: Chip 8 V0-VF variables
FFF0-FFF3: Graphic scratch area
FFF9: Keyboard code
FFFA: Chip 8 counter, counting down to 0 from value as set by Chip 8 instruction

Differences in Chip 8 instruction set

	Original Chip 8	VIP 2K Chip 8
User space	0x200 – 0xEFF	0x8000 – 0x8FFF, starting address is at 0x8200
00aa, SYS 0aa		Call 1802 system routine at 70aa
0aaa, SYS aaa	Call 1802 system routine at aaa	Call 1802 system routine at 8aaa (aaa > 0FF)
1aaa, JP aaa	Jump to address aaa	Jump to address 8aaa
2aaa, CALL aaa	Call subroutine at aaa	Call subroutine at 8aaa
Aaaa, LD I, aaa	I = aaa	I = 8aaa
Baaa, JP V0, aaa	Jump to address aaa + V0	Jump to address 8aaa + V0
Fx18, LD ST, Vx	Soundtimer = Vx	NOP

A FURTHER EXPLAINATION OF THE MONITOR COMMAND FORMATS

In all input commands, the leading zeros do not need to be inserted. For example, the following two command inputs are the same.

```
M0100 0005<CR>
M100 5<CR>
```

aaaa, bbbb, cccc Means that the monitor will see/allow only up to the last 4 characters inputted. For example the command value of 01234 and 821234 will be understood as a hexadecimal value of 1234 being inputted, which are only up to the last 4 characters inputted. If an inputted value is less than 4 characters long the monitor will add leading zeros to form the 4 digit value

dd Means that the monitor will see/allow only up to the last 2 characters inputted. For example the value of 234 and 34 will be understood as a hexadecimal value of 34 being inputted, which are only up to the last 2 characters inputted. An inputted value of 1 character the monitor will add a leading zero to form the 2 digit value (“01” in this case).

<CR> Means to press the "NL" key on the VIP2K keyboard
“ESC” key on the VIP2K is by pressing the Control and Number 1 at the same time.

While at the Command Prompt, all inputted characters are echoed to the display

The following are the only keys that are understood by this Monitor, “0" thru "9", "A" thru "F", Escape <ESC>, Space <SP>, Return <CR> keys, and Menu commands (M,W,T,R,V,D,B,H,A,C,I) while all other keys are ignored. NOTE: The monitor does not recognize the backspace key (BS) code.

Software Warranty

The Monitor Program software is a "work" in progress. Therefore there is NO warranty what-so-ever. I have personally run it on Lee Hart's VIP2K Membership Card Rev. A without any errors.

Enjoy,
Chuck Yakym