

Last Updated: August 6, 2017

Edited by: Kevin Dezuzio

Membership Card Serial Monitor Program Ver. 2.0

NOTE: Tiny Basic by Tom Pittman Copyright 1982

NOTE: RCA Basic 3 by Ron Cenker Copyright 1981

NOTE: The 1802 Membership Card Kit by Lee Hart, Copyrighted 2006

NOTE: Membership Card Serial Monitor Program by Charles Yakym Copyright 2015

NOTE: This Software was written to be used on Lee Hart's Membership Card Rev. G. or above. The Membership Card Kit can be purchased by contacting Lee at [leeahart@earthlink.net](mailto:leeahart@earthlink.net)

Please free feel to contact me if you have any problems and/or suggestions on how to use or improve this program. If you use this software and find any changes that this user manual needs or should be changed please feel free to contact me at [The--Eagle@sbcglobal.net](mailto:The--Eagle@sbcglobal.net) or at [The-Eagle@att.net](mailto:The-Eagle@att.net).

NOTE: This program has to be ran out of ROM memory with RAM in the lower 32K.

The Membership Card Serial Monitor program Ver. 2.0 once into a EPROM or ROM chip. Install the EPROM/ROM into U2 memory and you must have additional up to 32K RAM memory chip installed at U8 then run it by loading in a jump to address 8000h (C0 80 00). Make sure that the P2 and P3 memory jumpers on the Membership Card matches your EPROM/ROM type installed.

NOTE: Tiny Basic workspace is at 0000H and the user program begins at 0100H

#### INSTRUCTIONS ON HOW TO RUN THE MEMBERSHIP CARD SERIAL MONITOR PROGRAM VER 2.0

1. Load the following into memory address 0000H "C0 80 00"

2. Reset and Restart your Membership Card.

3. Now hit the ENTER key from your terminal program.

NOTE - You now should be seeing the Membership Card Serial Monitor Command Prompt.

NOTE: ALL keyboard depressions from your terminal/program are echoed to the LED's (OUT 4) on the Membership Card's Front Panel.

NOTE: ALL commands and Hex values in the Monitor are capitalized.

Now you should see "Membership Card's Serial Monitor Ver. 2.0 <CR> Enter "H" for Help". If you press an "H" then you will be viewing the commands and formats for each of them that are available while using this Monitor program.

The following commands are available, "M" reads from memory, "W" writes to memory, "T" transfers (copies) bytes located at a memory location to another, "R" Executes (RUNs)a program starting at the inputted address (PC=0, Q=0, X=0), "V" view 1802 registers, "S" save user program, "L"

load user program, "D" disassembler, "B" Tiny Basic, "C" save a Tiny Basic Program, "I" load a Tiny Basic Program, "N" RCA Basic3.

NOTE: Important things to remember when running this software are below.

Monitor Program Internal Serial I/O Routines:

I tested this Monitor Program using the following baud rates 300 8N1, 1200 8N1, 2400 8N1, 4800 8N1 and 9600 8N1. If you see garbled characters on your terminal screen, reduce your baud rate. NOTE: It Auto-Detects baud rate from 300 to 4800 baud. If the baud rate is any faster than 4800 then the Monitor Program will ASSUME that it is 9600 baud and that your Membership Card is running at a nominal clock speed of 1.8 Mhz.

NOTE: At all baud rates, the Monitor Program will automatically adjust itself internally to accommodate NORMAL or INVERTED RS-232 logic configuration that you are using to interface your PC to the Membership Card.

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 W1000 11 22 33<ESC> will only WRITE the data 11 and 22 into location starting at address 1000, 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 M1000b 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 either press the "Input Key" (S11) on the Membership's Card front panel or hit any key on your terminal (software or hardware) keyboard, if you are using a key on/in your terminal program then you might have to press it a few times (or press and hold). This is because the serial input is checked part of the time.

When using the "W" (write memory function) an input of W1000<CR> will result in a "Function Aborted" (No data to write). a W1000 11<CR> will write a value of 11 into location 1000. Now a W1000 11 22 33<ESC>, this will only write the 11 data into location 1000 and the data 22 into location 1001. 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 T1000 2000 0100<CR>, will transfer the memory from 1000 to 2000 for 0100 Hex bytes (1000h thru 10FFh). 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, R1000<CR> starts executing the code loaded into location 1000 with the Q line reset, X =0 and the

program counter is set to R0, which would contain the start address of 1000. The CDP1802 registers will be preloaded with the values displayed using the "V"iew register command with the exception of R0 and R1. R0 is set by the "R"un command while the R1 register is set to the re-entry point of the monitor program for a software interrupt via the D1 opcode. Register values can be changed by using the "W"rite command. (A description of this follows below.) You can change R0 and R1 but they will be preloaded with the values stated below.

R0 = users inputted address as defined by the "R"un command.

R1 = a value of 9615H, this address in R1 is the re-entry point back into the Monitor Program. NOTE: If you decide to change the value of R1 while running your program and you wish to return to the monitor, then instead of using the D1 opcode to return, you MUST make a jump to address 9615H (C0 96 15) to return if R1 is not restored.

Restrictions for your program using Break Points:

Register R4 and R5 can NOT be used as your program counter upon re-entry

Register R2 must point to RAM that has 5 available memory locations for storage upon returning.

Register R2 is used as a stack pointer upon return.

Register R2, can NOT point to a location that resides within the Monitor Program nor can it point to a memory location that is within the 64 bytes of the physical top of memory in your system.

NOTE: Using the MCSMP20 Program resides from 8000h thru FFFFh in EPROM/ROM.

NOTE: The last 64 bytes of the top of physical RAM memory is reserved by the monitor for its scratchpad area. If your program changes any value stored other than the register values in this area then the Monitor Program may crash.

NOTE: If you change the value in R1 then you must return via a jump as describe above.

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. NOTE: "XX" below refers to the last memory page of the top of your physical RAM memory address available.

For example:

Using a lower 32K RAM with an upper 32K EPROM/ROM then "XX" equals 7Fh

Address	Register
XXFF	D
XXFE	T
XXFD	X
XXFC	P
XXDB	DF
XXDC-XXDD	R0
XXDE-XXDF	R1
XXE0-XXE1	R2
XXE2-XXE3	R3
XXE4-XXE5	R4
XXE6-XXE7	R5
XXE8-XXE9	R6
XXEA-XXEB	R7

XXEC-XXED	R8
XXEE-XXEF	R9
XXF0-XXF1	RA
XXF2-XXF3	RB
XXF4-XXF5	RC
XXF6-XXF7	RD
XXF8-XXF9	RE
XXFA-XXFB	RF
HIGH-LOW	register

#### Inserting Breakpoints:

Break Points are a software interrupt set by the "D1" (SEP R1) opcode being inserted into your program. Upon executing the D1 opcode your program will return to the Monitor Program and you can then view the registers by using the "V"iew command. Remember that if your program changes R1, then you must jump back to the re-entry point (9615H).

#### Sample test program:

```
F8 12 B3 F8 34 A3 F8 40 B2 F8 00 A2 F8 80 FE D1
```

This sample program will load R3 with a value of 1234h then load R2 with a stack pointer value of 4000h then sets the DF register to equal 1 then returns to the Monitor Program via the D1 opcode. You can now use the "V"iew registers to see that this change has been made. You will notice that when using the "V"iew command that it will always display the following values: T=21h, X=2, and P=1. However, if you replace the D1 opcode with a direct jump (C0 96 15) then the P register will reflect your program counter register correctly. X remains R2 no matter which way you return to the Monitor Program. NOTE: The P register is labeled PC (Program Counter) in the "V"iew command.

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 was started from its base address (x000h), then register RE will always show the baud rate delay value plus the RS-232 baud rate info and polarity that the user is using. If this Monitor is re-entered via an software interrupt from a user program called by the "R" command it will display the correct values of all registers. See the "R"un command for more information.

When using the "L" (LOAD) user program command, It will display the following message "Ready to LOAD program". At this moment IF you decide not to load a program, you can just press the <ESC> Escape key to abort this function otherwise you MUST press the ENTER/RETURN key (<CR>) to start the load process. 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, 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.

When using the "I" load a Tiny Basic program command , it works just like the "L" (LOAD) command except that the file must be in ASCII form and NOT I8HEX format.

NOTE: Prior to using this command, you MUST have "Cold" started Tiny Basic then returned back to the Monitor Program. This will load up the 256 byte Tiny Basic workspace needed for this function to

work correctly. This function needs to see a non-corrupted Tiny Basic workspace memory page so it knows where to load your Basic program in at.

NOTE: Any Tiny Basic program loaded in the Monitor Program MUST have the last line in the Tiny Basic program file a line number of 0. See below

Tiny Basic Program listing:

```
10 LET A=0
20 LET B=B+A
30 PRINT B
40 LET A=A+1
50 GOTO 20
60 END
0
```

The last line number of zero tells the Monitor Program that the end of the file has been reached.

If there is no line number 0 then the Monitor Program will hang and will need to be restarted.

NOTE: When you save a Tiny Basic program from the Monitor Program it will automatically insert a last line with a line number of zero for you.

NOTE: If you plan on "loading" Intel Hex I8HEX format files or Tiny Basic programs into your system, 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 the Monitor Program says "File Loaded Successfully" after the "I" command, doesn't mean that the file was loaded "correctly". After you use the "I" to load a Tiny Basic Program, you should look (list) the program to make sure it is all correct BEFORE running it. The "File Loaded Successfully" means that the monitor program has loaded it and understood the last line number of zero. IF your program is not correct, then increase the "delay" and try reloading the Tiny Basic program.

When using the "S" (SAVE) user program command, It will display the following message "Ready to SAVE program". At this moment IF you decide not to SAVE a program, you can just press the <ESC> Escape key WITHIN about 15 seconds or less to abort this function. Once the file transfer has begun, 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, this monitor should display "File Saved Successfully". Be sure to end the capture and truncate any unnecessary text at the end of the capture file such as the "File Saved Successfully" text.

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.

Here is a detailed explanation of the Save command.

After you have selected the "Save function by pressing "S", the output LEDs on your system should be displaying "FF". This will be displayed for about 15 seconds. During this time you can do one of two things, you can abort this function by pressing the <ESC> Escape key or use this time to set up your terminal program to "Capture" a file from the serial port. If you fail to do anything during these 15 seconds, then your terminal program will display the data sent to it by the Save command. This data displayed on your screen should be considered unusable and meaningless.

If you have told your terminal program to capture the data, then after the first 15 seconds your system LEDs will start displaying the binary form of the characters sent to your terminal program to be captured.

After the Save command is done sending your program, the output LED's on your system will start blinking from "FF" (all ON) and then to "00" (all OFF). This is to let you know that the file has been sent correctly and that you should tell your terminal program to stop capturing the data. Once you have done that then just press any key on your terminal to return back to the Serial Monitor's command prompt.

When using the "C" save Tiny Basic user program command, like the "S" command, it will display the following message "Ready to SAVE program". At this moment IF you decide not to SAVE a program, you can just press the <ESC> Escape key WITHIN about 15 seconds or less to abort this function. Once the file transfer has begun, 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 output LED's on your system will start blinking from "FF" (all ON) and then to "00" (all OFF). This is to let you know that the file has been sent correctly and that you should tell your terminal program to stop capturing. Once you stop the capture, hit the <CR> Return key then this monitor should display "File Saved Successfully". The Tiny Basic user program is saved as an ASCII only text file and not an I8HEX 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 address 4000h, 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.

```
4000 7B SEQ
4001 F810 LDI 10
4003 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. NOTE: If you want to abort the disassembler command while it is displaying the disassembly contents you MUST either press the "Input Key (S11) on the Membership's Card front panel or hit any key on your terminal keyboard. If you are using a key on/in your terminal program then you might have to press it a few times, this is because of the limited timing of when the serial input is checked.

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 "D0000 " to debug from address 0000. (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 to continue. After the disassembly is displayed the disassembler is waiting for one of two things to happen. If you hit the <ESC> Escape key the disassembler is aborted and returns you back to the monitor's main command prompt while any other key depression 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 but they are displayed on the Membership Card's Front panel LED's.

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.

D9EA0 9FC5<CR>

RCA BASIC3

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

TINY BASIC

To enter Tiny Basic, enter the "B" command. The monitor program will prompt you if you want to do a Cold or Warm start of Tiny Basic. At this point; if you decide not to enter Tiny Basic, just press the <ESC> Escape key to abort this function.

NOTE: All Tiny Basic commands are in Capital letters only.

Cold Start

Upon entry into basic, it will reset all of its internal pointers and size memory to make all RAM memory available for use for your Basic program, except for the lowest 256 bytes which are reserved for Tiny Basic workspace area.

NOTE: You must Cold Start Tiny Basic BEFORE you use the "C" save or "I" load a Tiny Basic user program into or from the Monitor Program.

#### Warm Start

It is assumed that you already ran a cold start initialization and have loaded a Basic program into memory that is ready to be ran or edited.

NOTE: If you are in the monitor program and you used the "C" save your Tiny Basic program, then you can use the "I" load command to load your saved Tiny Basic then right afterwards you can enter Tiny Basic using the Warm start and your program is ready to run.

#### BREAK detection

While running a Tiny Basic program, you can "BREAK" out of its currently execution by one of two means.

1. By rapidly/repeating a key input from your terminal program or
2. By depressing the Input button on the Membership Card front panel.

#### The USR function within Tiny Basic

There are 8 embedded USR functions within Tiny Basic, they are as follows:

##### USR(-32768) - EXIT AND COLD START FUNCTION

Format: variable=USR(-32768)

Example program:

```
10 A=USR(-32768)
```

NOTE: This exit Tiny Basic and does a Cold start of the Membership Card Serial Monitor Program. Upon re-entry into the monitor program you MUST depress the <CR> Return key to reset the monitor program.

##### USR(-32765) - EXIT FUNCTION

Format: variable=USR(-32765)

Example program:

```
10 A=USR(-32765)
```

NOTE: This exit Tiny Basic and turns control back to the Membership Card Serial Monitor Program and functions as a warm start of the Monitor.

##### USR(-32762) - INPUT FUNCTION

Format: variable=USR(-32762)

The "variable" now contains the decimal value of the key pressed

NOTE: The input routine will automatically display the key pressed as well.

Example program:

```
10 LET A=USR(-32762)
```

```
20 PRINT A
```

```
30 IF A=13 THEN GOTO 50
```

```
40 GOTO 10
```

```
50 END
```



If you run the above program and press just the "A" key, then Tiny Basic will appear to display: A65. The "A" displayed is a function of the Input routine and NOT Tiny Basic. Tiny Basic responds by displaying only the "65".

#### USR(-32759) - OUTPUT FUNCTION

Format: variable=USR(-32759,N)

Where "N" equal value to print

Example program:

```
10 B=32
20 A=USR(-32759,B)
30 LET B=B+1
40 IF B=128 THEN GOTO 60
50 GOTO 20
60 END
```

NOTE: This program only prints the available ASCII characters

#### USR(-32748) - PEEK FUNCTION

Format: variable=USR(-32748,N)

The "variable" now contains the decimal value of the BYTE stored at address N

Example Program:

```
10 B=0
20 A=USR(-32748,B)
30 PRINT A
40 END
```

#### USR(-32744) - POKE FUNCTION

Format: variable=USR(-32744,address,N)

Where "address" equals decimal value of the address to store "N" at.

Example Program:

```
10 B=4096
20 c=32
30 A=USR(-32744,B,c)
40 END
```

NOTE: This program will load a value of 32 into location 1000H

#### USR(-32736) - MEMBERSHIP CARD FRONT PANEL LED OUTPUT FUNCTION

Format: variable=USR(-32736,N)

Where "N" equal value to print

Example program:

```
10 B=255
20 A=USR(-32736,B)
30 END
```

NOTE: This will turn on all the front panel LED's

USR(-32733) - MEMBERSHIP CARD FRONT PANEL SWITCHES READ FUNCTION - 8000h Monitor users must use USR(-32733)

Format: variable=USR(-32733)

Example program:

```
10 A=USR(-32733)
```

```
20 PRINT A
```

```
30 END
```

NOTE: This will display the decimal value of the Binary value of the switch settings on the Membership Card Front Panel

NOTE: Some parts of Tiny Basic and this Monitor also scans the (S11) Input button (/EF4 line) on the Front Panel of the Membership Card along with the /EF3 line (serial input line) to abort some commands while they are being executed.

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>
```

#### A FURTHER EXPLANATION OF THE COMMAND FORMATS

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 "Return or Enter" key on your keyboard

ALL inputs must be in CAPITAL letters; lower case letters are ignored.

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

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,S,L,T,R,V,D,B,C,I,H,N) while all other keys are ignored.

NOTE: The following doesn't apply when using "BreakPoints" in your program to return to the Monitor Program.

If you use the "R"un command in this Monitor to run your own program and you want to re-enter this Monitor (at its base address) from your program, then you MUST follow these requirements. Upon re-entry at its base address (8000h), You MUST hit the Return <CR> key on your keyboard to reset the Bit timing value. After I re-structure the internal layout of Monitor Program I will add a section on how to use some of its internal routines from your program.

### 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 Membership Card Rev. G, Netronics ELF II system, and my 1802 (64K) test platform without any errors.

Enjoy,  
Chuck Yakym

P.s. I am currently using this monitor program to allow me to use my PC laptop to act as a display and keyboard for the Membership Card via RealTerm (terminal program). Realterm is FREE and available at the following website: <http://sourceforge.net/projects/realterm>. I also use a TRENDnet TU-S9 (USB to Serial adapter which is available at <http://www.amazon.com>) to connect via the RS-232 circuit on Membership Card Rev. G.