## Backing up and restoring the HP15c user memory (with file)

Make sure you have the firmware dated 2024-06-03 and with checksum 0A0Ah. If you have a prior firmware, you need to get the firmware file from this location, and install it following the instructions listed here.

Copy the file VoyagerSave.exe to your computer.

Connect the computer with the programming cable.

Once you have the right firmware, switch the calculator off, press f & ENTER simultaneously, and while keeping them pressed, press "on".

The calculator is now set in USB HID mode. It can be seen in your computer device manager as "Voyager".

(To exit the HID mode you can at any time press "on" to switch the calculator off and come back to standard mode)

Open the command line in your computer (by running cmd in the search line in your windows menu). Linux may need other procedure. For Mac I am using Parallels and a Win11 virtual installation.

Navigate through folders to the folder where VoyagerSave is located.

When executing VoyagerSave without parameters, it gives help on its use:

```
C:\Users\josegonzalezdivasson\Documents>VoyagerSave
VoyagerSave -s file: Save HP15C/HP16C calc structure to file
VoyagerSave -r file: restore calc structure from file
```

To back up your calculator, write **VoyagerSave -s filename** and press enter, where "filename" is the name you want to give to the backup file. The system backs up the user memory area and stores it in the file. You will save the user memory area of the mode you are currently in. This is interesting to remember when you are moving from HP15c to HP16c and viceversa, to avoid losing your keyed-in programs. To restore from a previous backup, write **VoyagerSave -r filename**, where "filename" is the name you gave to the back up in point 9.

Press "on" to exit from USB HID mode.

Disconnect the programming cable, pushing gently towards the center on the connecting head side wings. Remember to close and secure the back door!

Of course, you can have as many different backups as needed. A good policy is creating different user profiles, depending on the situation and application. These can be thought as "software pacs", since you can reconfigure your calculator for specific applications, and be able to call them to memory when you need these programs. We plan to create some of these software pacs.

## Structure of the back-up file.

The back-up files are just dumps of the user memory area, which is 2048 bytes long, and is organized as shown below. Registers are 8 bytes long, so there are 256 8-byte registers:

Mantissa Random number

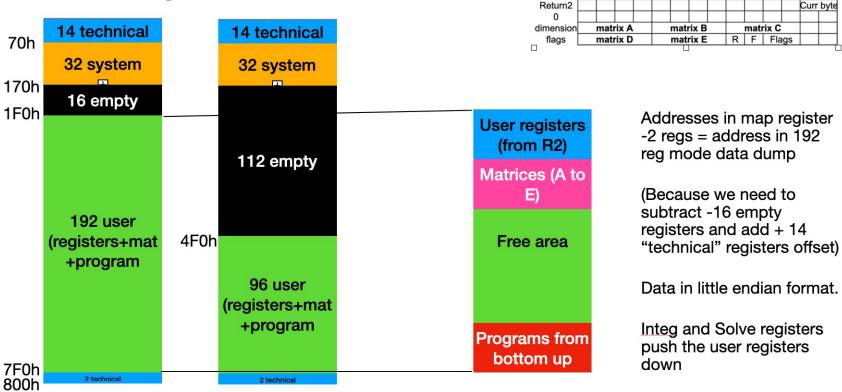
last line

R1 I Last X

ran

map
 Return1

## Data dump format In 192 and 96 register mode.



There are two areas of unknown content so far, the **blue** areas in the picture: 14 registers at the beginning and 2 registers at the end. These were not present in the original memory of the calculator, and appear only in the emulator.

The original HP15c had 7-bytes registers, while the file registers are 8-bytes long. This means that there is always an empty byte at the end of each HP15c register. (this does not apply to the blue "technical" registers).

The discussion below can be better understood by reading Syhthetic Methods on the HP15c. While the HP15c CE does not accept synthetic methods, the discoveries found using them on the original calculator give us information about how the user memory is distributed - and this is applicable here. The 32 first registers are the System Registers. Its distribution is the following:

- The first column is the system register number;
- the second column is its location in the file;

- the third is its known use;
- the 4th and 5th are the two nibbles for the padding byte;
- and the following are the 14 nibbles of the 7 bytes of the original HP15c registers, starting with the sign, the mantissa, the sign of the exponent and the two exponent numbers.

For our purposes, it is important to see that the stack is there, and also registers 0 and 1 (and this is the reason that they cannot be allocated to programs, too. It also may imply that operations done to these registers might be slightly faster than with other registers, where we need to have memory map access):

Hex offset	Byte in file		P1	P2	S	1	2	3	4	5	6	7	8	9	0	S		
00	70	у																
01	78	z																
02	80	t																
03	88	Pointer																
04	90	Display																
05	98	?																
06	A0	?																
07	A8	Exponent																
08	B0	0																
09	B8	Control L																
0A	C0	Control R																
0B	C8	0																
0C	D0	0																
0D	D8	0																
0E	E0	0																
0F	E8	0																
10	F0	R0																
11	F8	R1																
12	100	- 1																
13	108	Last X																
14	110	ran		Run) Ma					ntissa	Rar	ndom	num	ber			curr.		
15	118	map		Data		mA		mB		mC		mD		m	ıΕ			
16	120	Return1																
17	128	Return2																
18	130	0																
19	138	dimension				mat	rix A	3	matrix B					matrix C				

	1	.00000000	 (00000				(0.0000				(0)0000			
1A	140	flags	matrix D			matrix E				R	F	Flags		
1B	148	0												
1C	150	0												
1D	158	0												
1E	160	0												
1F	168	0												

VoyagerSave