RomWBW ROM Applications

Phillip Summers

Sunday 29 August 2021

Contents

Summary

RomWBW includes a small selection of built in utilities and programming languages.

ROMWBW Monitor

The Monitor program is a low level utility that can be used for testing and programming. It allows programs to be entered, memory to be examined, and input/output devices to be read or written to.

It’s key advantage is that is available at boot up.

Its key disadvantages are that code cannot be entered in assembly language and there is no ability to save to memory devices.

The available memory area for programming is `0200-EDFFh`. The following areas are reserved:

Memory Area	Function
`0000-00FFh`	Jump and restart (RST) vectors
`0100-01FFh`	HBIOS configuration block
`FE00-FFFFh`	HBIOS proxy

Commands can be entered at the command prompt > Automatic case conversion takes place on command entry and all arguments are expected to be in hex format.

The current memory bank in low memory is displayed before the prompt i.e.:

```
8E>
```

The Monitor allows access to all memory locations but ROM and Flash memory cannot be written to. Memory outside the normal address range can be accessed using the B command. The first 256 bytes `0000-01FF` is critical for the HBIOS operation. Changing banks may make this information inaccessible.

Refer to the RomWBW Architecture manual for details memory banking.

A quick guide to using the Monitor program follows:

? - Displays a summary of available commands.

```
Monitor Commands (all values in hex):
B                    - Boot system
D xxxx yyyy          - Dump memory from xxxx to yyyy
F xxxx yyyy zz       - Fill memory from xxxx to yyyy with zz
H                    - Halt system
I xxxx               - Input from port xxxx
K                    - Keyboard echo
L                    - Load Intel hex data
M xxxx yyyy zzzz     - Move memory block xxxx-yyyy to zzzz
O xxxx yy            - Output value yy to port xxxx
P xxxx               - Program RAM at address xxxx
R xxxx [[yy] [zzzz]] - Run code at address xxxx
                       Pass yy and zzzz to register A and BC
S xx                 - Set bank to xx
X                    - Exit monitor
```

Cold Boot

B - Performs a cold boot of the ROMWBW system. A complete re-initialization of the system is performed and the system returns to the Boot Loader prompt.

Dump Memory

D xxxx yyyy - Dump memory from hex location xxxx to yyyy on the screen as lines of 16 hexadecimal bytes with their ASCII equivalents (if within a set range, else a '.' is printed). A good tool to see where code is located, check for version id, obtain details for chip configurations and execution paths.

Examples: `D 100 1FF`

```
0100: 10 0B 01 5A 33 45 4E 56 01 00 00 2A 06 00 F9 11  ...Z3ENV...*..ù.
0110: DE 38 37 ED 52 4D 44 0B 6B 62 13 36 00 ED B0 21  Þ87íRMD.kb.6.í°!
```

```
0120: 7D 32 E5 21 80 00 4E 23 06 00 09 36 00 21 81 00  }2å!..N#...6.!..
0130: E5 CD 6C 1F C1 C1 E5 2A C9 8C E5 CD 45 05 E5 CD  åÍl.ÁÁå*É.åÍE.åÍ
0140: 59 1F C3 00 00 C3 AE 01 C3 51 04 C3 4C 02 C3 57  Y.Ã..î.ÃQ.ÃL.ÃW
0150: 02 C3 64 02 C3 75 02 C3 88 02 C3 B2 03 C3 0D 04  .Ãd.Ãu.Ã..ò.Ã..
0160: C3 19 04 C3 22 04 C3 2A 04 C3 35 04 C3 40 04 C3  Ã..Ã".Ã*.Ã5.Ã@.Ã
0170: 48 04 C3 50 04 C3 50 04 C3 50 04 C3 8F 02 C3 93  H.ÃP.ÃP.ÃP.Ã..Ã.
0180: 02 C3 94 02 C3 95 02 C3 85 04 C3 C7 04 C3 D1 01  .Ã..Ã..Ã..ÃÇ.ÃÑ.
0190: C3 48 02 C3 E7 04 C3 56 03 C3 D0 01 C3 D0 01 C3  ÃH.Ãç.ÃV.ÃÐ.ÃÐ.Ã
01A0: D0 01 C3 D0 01 C3 D0 01 C3 D0 01 01 02 01 CD 6B  Ð.ÃÐ.ÃÐ.ÃÐ....Ík
01B0: 04 54 68 69 73 20 66 75 6E 63 74 69 6F 6E 20 6E  .This function n
01C0: 6F 74 20 73 75 70 70 6F 72 74 65 64 2E 0D 0A 00  ot supported....
01D0: C9 3E FF 32 3C 00 3A 5D 00 FE 20 28 14 D6 30 32  É>ÿ2<.:].þ (.Ö02
01E0: AB 01 32 AD 01 3A 5E 00 FE 20 28 05 D6 30 32 AC  «.2-
.:^.þ (.Ö02¬
01F0: 01 C5 01 F0 F8 CF E5 26 00 0E 0A CD 39 02 7D 3C  .Å.ðøÏå&...Í9.}<
```

Fill Memory

F xxxx yyyy zz - Fill memory from hex xxxx to yyyy with a single value of zz over the full range. The Dump command can be used to confirm that the Fill completed as expected. A good way to zero out memory areas before writing machine data for debug purposes.

Halt System

H - Halt system. A Z80 HALT instruction is executed. The system remains in the halt state until the system is physically rebooted. Interrupts will not restart the system. On systems that support a HALT status LED, the LED will be illuminated.

Input from port

I xxxx - Input data from port xxxx and display to the screen. This command is used to read values from hardware I/O ports and display the contents in hexadecimal.

Keyboard Echo

K - Echo any key-presses from the terminal. Press ‘ESC’ key to quit. This facility provides that any key stroke sent to the computer will be echoed back to the terminal. File down loads will be echoed as well while this facility is ‘on’.

Load Hex format file into memory

L - Load a Intel Hex format file via the terminal program. The terminal emulator program should be configured to give a delay at the end of each line to allow the monitor enough time to parse the line and move the data to memory. Keep in mind that this will be a transient unless the system support backed memory. Saving to memory drive is not supported.

Move memory

M xxxx yyyy zzzz - Move hex memory block xxxx to yyyy to memory starting at hex location zzzz. Care should be taken to insure that there is enough memory at the destination so that code does not get over-written or memory wrapped around.

Output to port

O xxxx yy - Output data byte xx to port xxxx. This command is used to send hexadecimal values to hardware I/O ports to verify their operation and is the companion to the I operation. Use clip leaded LEDs to confirm the data written.

Program memory location

P xxxx - Program memory location xxxx. This routine will allow you to program a hexadecimal value into memory starting at location xxxx. Press ‘Enter’ on a blank line to return to the Monitor prompt.

Run program

R xxxx [[yy] [zzzz]] - Run program at location xxxx. If optional arguments yy and zzzz are entered they are loaded into the A and BC register respectively. The return address of the Monitor is saved on the stack so the program can return to the monitor. On return to the monitor, the contents of the A, HL, DE and BC registers are displayed.

NOTES:

The RTC utility on the CP/M ROM disk provides facilities to manipulate the Real Time Clock non-volatile Memory. Use the C or Z option from the Boot Loader to load CP/M and then run RTC to see the options list.

FORTH

BASIC

TastyBASIC

Play a Game

2048

2048 is a puzzle game that can be both mindless and challenging. It appears deceptively simple but failure can creep up on you suddenly.

It requires an ANSI/VT-100 compatible colour terminal to play.

2048 is like a sliding puzzle game except the puzzle tiles are numbers instead of pictures. Instead of moving a single tile all tiles are moved simultaneously in the same direction. Where two tiles of the same number collide, they are reduced to one tile with the combined value. After every move a new tile is added with a starting value of 2.

The goal is to create a tile of 2048 before all tile locations are occupied. Reaching the highest points score, which is the sum of all the tiles is a secondary goal. The game will automatically end when there are no more possible moves.

Play consists of entering a direction to move. Directions can be entered using any of three different keyboard direction sets.

```
Direction | Keys
----------|----------
Up        | w ^E 8
Down      | s ^X 2
Left      | a ^S 4
```

```
Right     | d ^D 6
```

The puzzle board is a 4x4 grid. At start, the grid will be populated with two 2 tiles. An example game sequence is shown below with new tiles to the game shown in brackets.

```
Start              Move 1 - Up        Move 2 - Left      Move 3 - Left
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
|   |   |   |(2)|  |   |   |   | 4 |  | 4 |   |   |   |  | 4 |   |   |   |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
|   |   |   |   |  |   |   |   |   |  |   |   |   |(4)|  | 4 |   |   |   |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
|   |   |   |(2)|  |   |   |   |   |  |   |   |   |   |  |   |   |   |   |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
|   |   |   |   |  |   |   |(2)|   |  | 2 |   |   |   |  | 2 |   |(2)|   |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+

Move 4 - Left      Move 5 - Up        Move 6 - Right     Move 7 - Up
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
| 4 |   |   |   |  | 8 |   |   | 4 |  |   |   | 8 | 4 |  |   |   | 8 | 8 |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
| 4 |   |   |(4)|  | 4 |   |   |   |  |   |   |   | 4 |  |   |   |   | 2 |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
|   |   |   |   |  |   |   |   |   |  |   |   |   |   |  |   |   |   |   |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
| 4 |   |   |   |  |(2)|   |   |   |  |(2)|   |   | 2 |  |(2)|   |   |   |
+---+---+---+---+  +---+---+---+---+  +---+---+---+---+  +---+---+---+---+
```

This is how I lost this game:

```
+---+---+---+---+
| 4 | 2 | 16| 4 |
+---+---+---+---+
| 32| 64| 8 | 2 |
+---+---+---+---+
| 4 | 8 |128| 32|
+---+---+---+---+
|(2)| 16| 8 | 4 |
+---+---+---+---+
```

Press Q at any time to bring up the option to Quit or Restart the game.

Network Boot

ZModem Flash Update